1994-09

Mandatory security policy enforcement in commercial off the shelf database management system software : a comparative analysis of Informix On- Line/Secure and trusted ORACLE

Muschalek, Keith Edward

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MANDATORY SECURITY POLICY ENFORCEMENT IN COMMERCIAL OFF THE SHELF DATABASE MANAGEMENT SYSTEM SOFTWARE: A COMPARATIVE ANALYSIS OF INFORMIX ON-LINE/SECURE AND TRUSTED ORACLE

by

Keith E. Muschalek

September 1994

Co-Advisors: Cynthia Irvine
C. Thomas Wu

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1994	Master's Thesis

4. TITLE AND SUBTITLE
MANDATORY SECURITY POLICY ENFORCMENT IN COMMERCIAL OFF THE SHELF DATABASE MANAGEMENT SYSTEM SOFTWARE: A COMPARATIVE ANALYSIS OF INFORMIX-ONLINE/SECURE AND TRUSTED ORACLE (U)

5. FUNDING NUMBERS

6. AUTHOR(S)
Muschalek, Keith Edward

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective of this thesis is to analyze the mandatory access control (MAC) features of two commercial multilevel trusted database management systems (DBMS): Trusted ORACLE 7 and Informix-OnLine/Secure 5.0. We are attempting to determine how the problem of multilevel sharing of information is addressed in each multilevel secure DBMS.

Commercially available documentation is used to examine the mandatory access controls enforced on labeled subjects and labeled objects and to compare them to the Class B1 requirements for MAC and labeling set forth in the Trusted Computer System Evaluation Criteria (TCSEC). A decomposition of the TCSEC requirements for MAC and labeling is mapped to the DBMS documentation to determine if the Class B1 requirements are met by each DBMS. With the TCSEC mapping as a reference, the interface features in support of MAC are analyzed and compared between the products.

This analysis shows that each DBMS uses different schema objects and privilege sets to enforce its mandatory security policy. The MAC mechanism of each product is based on the Bell-LaPadula security model, extended to prohibit the writeup of data from lower level subjects to higher level objects. Each DBMS allows traditional trusted subjects to writedown data. When special privileges are granted to users, readups and writeups are permitted in both DBMSs.

14. SUBJECT TERMS
Database Security, Multilevel Secure Database Management Systems, B1 DBMS, TCSEC analysis, Database analysis, Database evaluations.

15. NUMBER OF PAGES
163

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

ABSTRACT

The objective of this thesis is to analyze the mandatory access control (MAC) features of two commercial multilevel trusted database management systems (DBMS): Trusted ORACLE 7 and Informix-OnLine/Secure 5.0. We are attempting to determine how the problem of multilevel sharing of information is addressed in each multilevel secure DBMS.

Commercially available documentation is used to examine the mandatory access controls enforced on labeled subjects and labeled objects and to compare them to the Class B1 requirements for MAC and labeling set forth in the Trusted Computer System Evaluation Criteria (TCSEC). A decomposition of the TCSEC requirements for MAC and labeling is mapped to the DBMS documentation to determine if the Class B1 requirements are met by each DBMS. With the TCSEC mapping as a reference, the interface features in support of MAC are analyzed and compared between the products.

This analysis shows that each DBMS uses different schema objects and privilege sets to enforce its mandatory security policy. The MAC mechanism of each product is based on the Bell-LaPadula security model, extended to prohibit the writeup of data from lower level subjects to higher level objects. Each DBMS allows traditional trusted subjects to writedown data. When special privileges are granted to users, readups and writeups are permitted in both DBMSs.

I. INTRODUCTION

The security of information within computer systems is a major issue for system automation professionals of the 1990's. The disclosure of sensitive information, the modification of valuable data files, and the disruption of service, by both authorized and unauthorized personnel, have plagued system administrators for many years. Today, with the advent of interactive network computing and the "information superhighway", information has to be protected more than ever.

Security issues have been a concern in the national security and defense establishments since the dawn of the computer age. National defense mandated major requirements for security in the development and acquisition of automated computer systems. Work by government personnel and defense contractors brought about the development of systematic criteria for measuring the effectiveness and trustworthiness of security mechanisms within computer systems. These "criteria" became the Trusted Computer Security Evaluation Criteria (TCSEC), commonly called the "Orange Book." The TCSEC is the metric by which the United States Government measures the security effectiveness of an automated computer system.[1]

This thesis will be an attempt to conduct a comparative analysis of selected security features of two leading U.S. database management system (DBMS) products, (Trusted ORACLE 7.0 and INFORMIX On-Line/Secure 5.0), against the TCSEC.

A method of analysis will be presented which is based on mapping decomposed TCSEC criteria (and interpretations to the Criteria) to the database through a detailed analysis of each product's documentation and users' manuals. This method could be applied to assist in the analysis of any DBMS software product. It should be noted, that this

1. Other countries, such as Canada and the European community have their own criteria for measuring the security effectiveness of computer systems.

protected from modification by others, and that the information or computer resources are always available to them. Simply put, computer security focuses on secrecy, integrity, and denial of service.

The Privacy Act of 1974 and the Computer Security Act of 1987, which mandated that information be protected in automated information systems, is a driving force for both government and private industry to secure the data contained within there computer systems.

1. Security Objectives

The following sections further define the notion of what computer security is by additionally defining the three components: secrecy, integrity, and denial of service.

a. Secrecy

The secrecy component of security has been a prime focus of U.S. Government funded programs since the early 1970's. [GASS88] The objective is to protect the secrecy of classified information and government secrets. Because of the U.S. government's profound interest in secrecy, this aspect of computer security is well researched and studied by computer scientists [AMOR94]. Secrecy is intended to prevent the "leakage" of information from authorized users to unauthorized users.

b. Integrity

The integrity component of security covers the unauthorized modification of information stored in computer systems. Only authorized users of the system with the proper access to information should be able to alter (i.e., write, delete, append) data within the computer. If any other users change the information, then an integrity violation has occurred.(Note that authorized users of the system can still make erroneous changes to information and this would not be a integrity violation.)

Integrity of data has been a primary issue in the commercial business environment, with secrecy taking a secondary role [GASS88][AMOR94]. Businesses were

cleaner more reliable architecture.[GASS88] Hardware security is included in a computer system's evaluation, and may only be a peripheral consideration during a DBMS evaluation (or other application evaluation). The hardware and the operating system provide the "platform" on which the DBMS, (as well as other application programs) operate on.

b. Operating System Security

The second layer of security in a secure computing system is found at the operating system (OS) level. Access to objects (i.e., information containers having labels) is a primary focus of OS security. Discretionary and nondiscretionary access controls are present in most secure operating systems. Some operating systems have been developed which utilize a security kernel, which gives high assurance that a particular security policy is enforced. Many operating system products have been evaluated by the NSA since 1982 (the year evaluations began) [CHOK92]. Due to the system architecture of current DBMS implementations, certain security features of the OS (discretionary access controls and mandatory access controls on files and directories, for example) have to be explored when analyzing a DBMS product.

c. Database Security

A database management system (DBMS) is a complex software system designed to manipulate, store, and "manage" large amounts of raw data. Today's DBMSs are complicated, consisting of tens to hundreds of thousands of lines of code. Part of their specification requirements mandate them to provide for security and integrity of data. These security features are in addition to the security features provided by the underlying operating system.

The chief security features of standard (i.e., untrusted) database systems are account creation, account privileges, stored procedures and views, and audit logs. The more sophisticated DBMS products provide these and other more sophisticated mechanisms, while some of the lower-end products (such as PC based DBMS products) provide little or no security features at all.

which underlies both DBMS's. Chapters V and VI discuss the general architecture and operations of Trusted ORACLE and On-Line/Secure, respectively. The method of security analysis is presented in Chapter VII, and the detailed analysis for Trusted ORACLE and Informix-OnLine/Secure are presented in Chapters VIII and IX, respectively. A comparative analysis is presented in Chapter X, and conclusions, recommendations and future research are addressed in Chapter XI.

E. SCOPE, LIMITATIONS, AND ASSUMPTIONS

All information for this thesis was gathered from the open literature, interviews, and marketing documents. No special agreements have been arranged with the vendors or the evaluators for the disclosure of information about their respective software products.

tables, etc.) at different classifications. A system which processes data at many different levels and whose users are also classified at many different levels is called a multilevel system (MLS). When a computer system must be responsible for access mediation, an evaluated system provides a level of assurance that access control policy is correctly enforced.

B. THE REFERENCE MONITOR CONCEPT

To address the multilevel sharing issue a new way of doing business was needed, so in the late 1960's serious work began to address this problem. Early experiences with computer security were characterized by "Tiger teams"[3] which went out and tried to penetrate a computer system's defense. Once the team penetrated the system controls and "broke into" the computer, another team of computer scientists would fix the holes discovered by the Tiger teams. This early method of computer security has been referred to as the "penetrate and patch" approach; systems were tested to uncover flaws, and the penetration paths uncovered were then patched.[NCSC92a] This process of discovering problems led to even more problems and soon a system became heavily patched with "spaghetti code" intended to prevent unauthorized users from entering the system without permission. This "penetrate and patch" methodology is unreliable because no one can decide when any more flaws exist. This was no way to establish that a system was secure. A more general approach was needed.

A research project performed on behalf of the DOD [ANDE72] produced the reference monitor concept in the early 1970's. In this concept of "a reference monitor which enforces the authorized access relationships between subjects and objects of a system" [DOD85], a fundamentally new approach to the multilevel sharing issue was formulated. The Anderson Report [ANDE72] described the architectural framework needed for dealing with the mediation of access in the face of potentially hostile users. [NCSC92a]

---

3. Tiger teams are teams of computer scientists who simulate adversaries or threats, and try to penetrate the security holes in a computer system. Primarily used in the DOD.

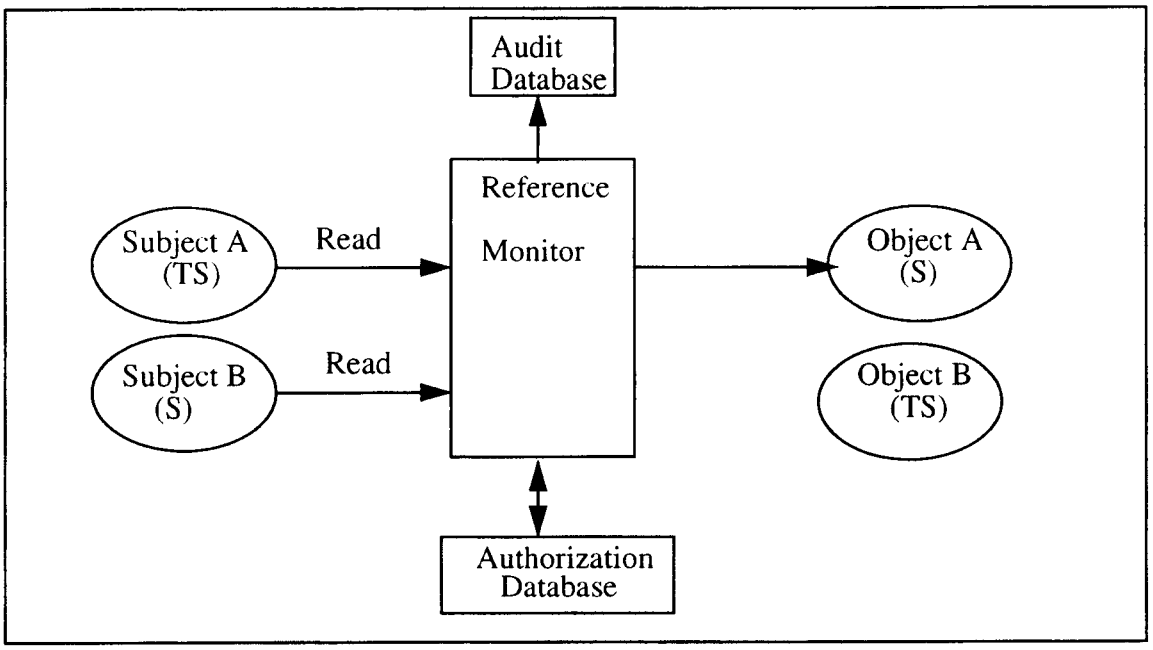

Figure 1: The Reference Monitor Concept

In Figure 1, Subject A is given permission to access (read) Object A. This is because Subject A's label (TOP SECRET) dominates (is greater than) Object A's label (SECRET). However, in the case of Subject B, its label (SECRET) is dominated by Object B's label (TOP SECRET) and is denied permission to access. A complete audit trail may be kept on all access attempts by writing to an audit database or audit file.

The reference monitor has become the general solution to the multilevel sharing problem. It is the most often used approach for building secure operating systems [GASS88] and represents an ideal approach for building access control features within trusted DBMSs. Before a discussion on how the reference monitor concept is implemented to control access to data and to resources (objects) by users (both authorized and unauthorized), the notion of a subject and an object is presented in the next section.

database objects, and object identification. This area is especially important in the area of DBMS evaluations and has been the topic of some discussion in the context of DBMS objects versus operating system objects [GRAU90].

An object is defined as a "passive entity that contains or receives information. "Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory trees, programs, bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, and network nodes."[DOD85]

An object can be thought of as a container, like a bucket, which can hold data. This bucket can be filled up (written to in the computer vernacular) and/or drained (read from) by the active processes (subjects, as previously discussed) in the system. However, this is a special bucket, when you drain it, the data inside does not really move at all, so no data is ever lost when the bucket is drained. In the technical sense, this object is a repository of data, which has an internal state that is changed (written) and/or observed (read) by the subjects of the computer system [NCSC89]. All state changes of the object are initiated by a set of well-defined operations that are available to the subjects[NCSC89]. One could call this category of objects "data objects", or one could categorize them as storage objects or named objects, depending upon how they are created and managed by the TCB. (See section below.)

1. Object Categories

Data objects can be thought of as containers which hold data and is a broad category in which to classify objects. One way to classify objects is by their physical properties, such as memory blocks or segments. Another useful way that we can classify objects is by the way in which subjects can access them [NCSC89]. In the context of DBMSs, if subjects can access objects through discretionary access controls, these objects are called "named objects." If subjects can access objects through mandatory access controls, then these objects are called "storage objects." [NCSC89]

be in a "privileged mode". A process operating in a privileged mode can access more of the memory address space (i.e., more objects), or it can invoke special system functions.

For example, the person who operates and maintains the computer system is called the system administrator. The system administrator usually requires special privileges to keep the computer system running correctly.

Another example of a special privilege is a system's backup and recovery program. The backup program will be allowed to bypass read restrictions on files so it can copy them to a magnetic tape or a floppy disk; the recovery program will be allowed to bypass write restrictions on files so it can write files and restore them to original form.[GASS88]

1. Least Privilege

This notion of different privileges given to the processes running in the system was inspired by the principle of least privilege. Least privilege is the concept that users (subjects)be granted only as many resources as they need to complete their job. The Orange Book defines least privilege explicitly as:

> ...that each subject in a system be granted the most restrictive set of privileges (or lowest clearance) needed for the performance of authorized tasks. "The application of this principle limits the damage that can result from accident, error, or unauthorized use.[DOD85]

The idea is to reduce the number of potential interactions between programs to the minimum amount needed for correct operation, so that if erroneous input is introduced by users or improper functions are called, the amount of damage to the system will be minimized. [SALT75]

2. Modes of Execution

The concept of modes of execution is crucial for the enforcement of least privilege. A mode of execution (or domain of execution) is the environment in which a process (subject) operates, and contains all those resources (objects) for which it has access. The Intel's iAPX x86 CPU, can operate in four modes. Assuming the programmers

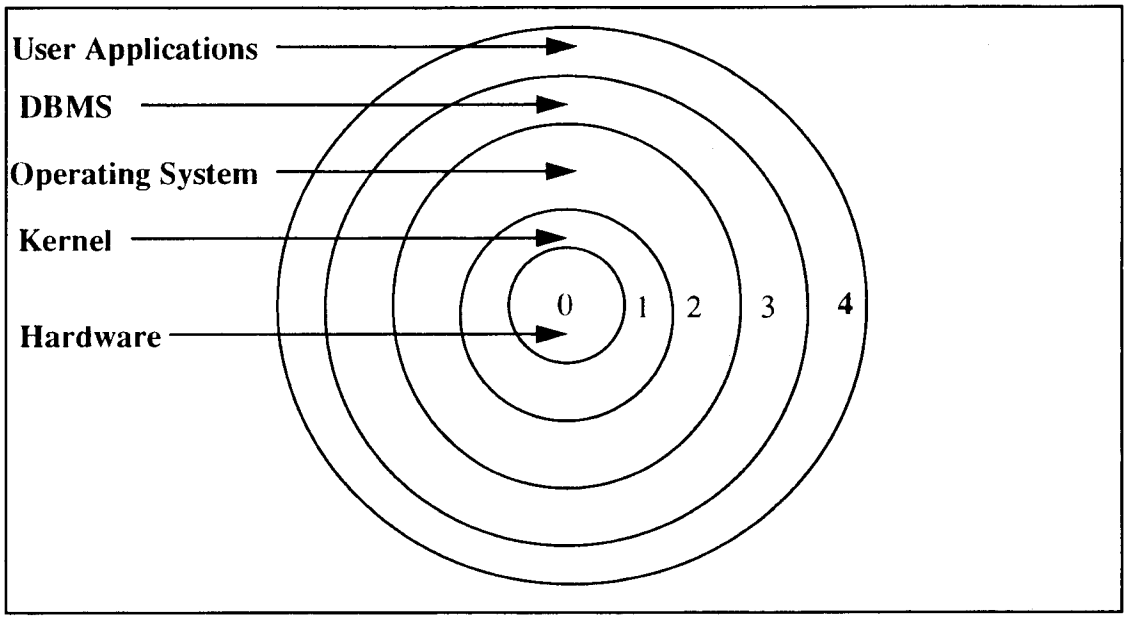

Figure 2: Hierarchial Domains or Rings

When a process executes, its respective subjects (a process may have more than one subject) operate within a predetermined ring or domain. A domain represents all the objects to which the subject has access (read or write). A subject's domain at any particular time might include a variety of programs, files, data segments, and I/O devices such as printers and terminals. Such a domain is shown in Figure 3.

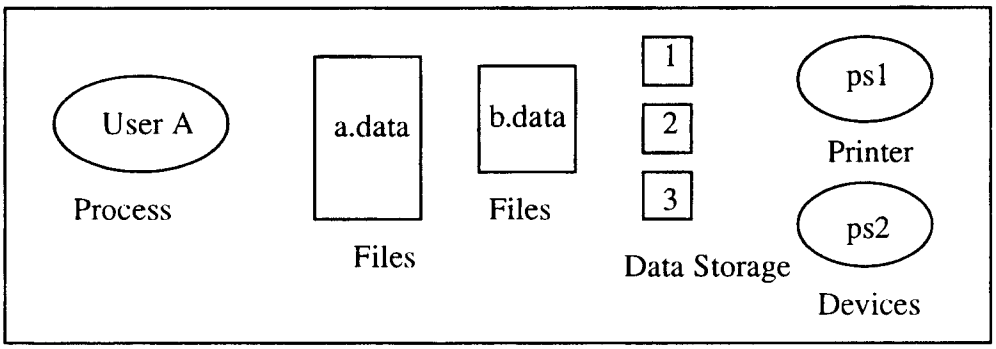

Figure 3: Domain of Execution

the set of rules, directives, and practices that regulate how an organization manages, protects, and distributes sensitive information.[DOD85]

The security policy describes every aspect of how information will be handled both inside the system and outside the system. Sensitive information is defined and could include everything from government TOP SECRET documents to a small company's proprietary business information or its personnel database. The security policy, once defined, is then translated into a security implementation within the computer system. The desired attributes of the system are eventually realized, in part, by the implementation of some specific set of mechanisms; functions which can be shown to provide the required attributes. [NCSC92a] The critical point is that one starts with a security policy (i.e., a high-level statement of the desired global properties or characteristics of the system), then proceeds through a number of refinement steps culminating with a set of specific implementations. [NCSC92a] See "SECURITY IMPLEMENTATIONS" on page 26.

1. Security Models

The security policy of a computing system can vary from short (i.e., no person outside the company should access this data) to extremely complex (e.g., U.S. Government TOP SECRET information systems). Formal security policies have been proposed and formulated into security models for several years. Both single-level and multi-level security models exist, but for the purposes of this thesis, only multi-level security models are of interest. In the multi-level world, multi-level security models, where many different objects and subjects of different classification are present, is our focus.

a. Military Security Model

We start with the military security model because most secure computer systems used by the defense and national security establishments are based on this model. This model also represents the base upon which many other important multilevel security models are built (i.e., the Bell-LaPadula Model).

information. These compartments are then used to enforce the need-to-know principle, so that users can obtain access only to information which is relevant to their jobs. Examples of compartments are TERRORISTS, NUCLEAR, and SPIES. A single piece of information would then be coded with zero or more compartments, depending on the categories to which the information applies (see Figure 5). For a person to gain access to a piece of information, he/she must possess all the compartments associated with the information, as well as a sensitivity classification that dominates the label of the information. For example, if Captain Smith wants TOP SECRET information that deals with nuclear spies and terrorists, he must have a security rating of at least TOP SECRET and compartment clearances for NUCLEAR, SPIES, and TERRORISTS. (See row 1 of Figure 5) Clearances in rows 2-5 (Figure 5) would not be adequate to gain the desired information.

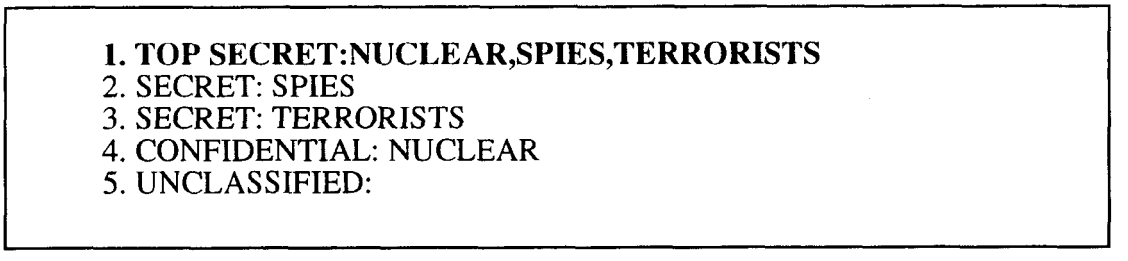

Figure 5: Military Security Labels

b. Bell-LaPadula Model

One of the best known and most popular multilevel security models is the Bell and LaPadula model developed and published by D. Bell and L. LaPadula in 1973. [BELL73] The Bell-LaPadula model (BLP) describes the allowable paths of access control in a secure system. The goal of the model is to identify allowable communication channels where it is important to maintain secrecy. (Note that this model does not preserve the integrity of the information).The model has been used to define the security requirements for systems concurrently handling data at different sensitivity levels. [PFLE89] This model

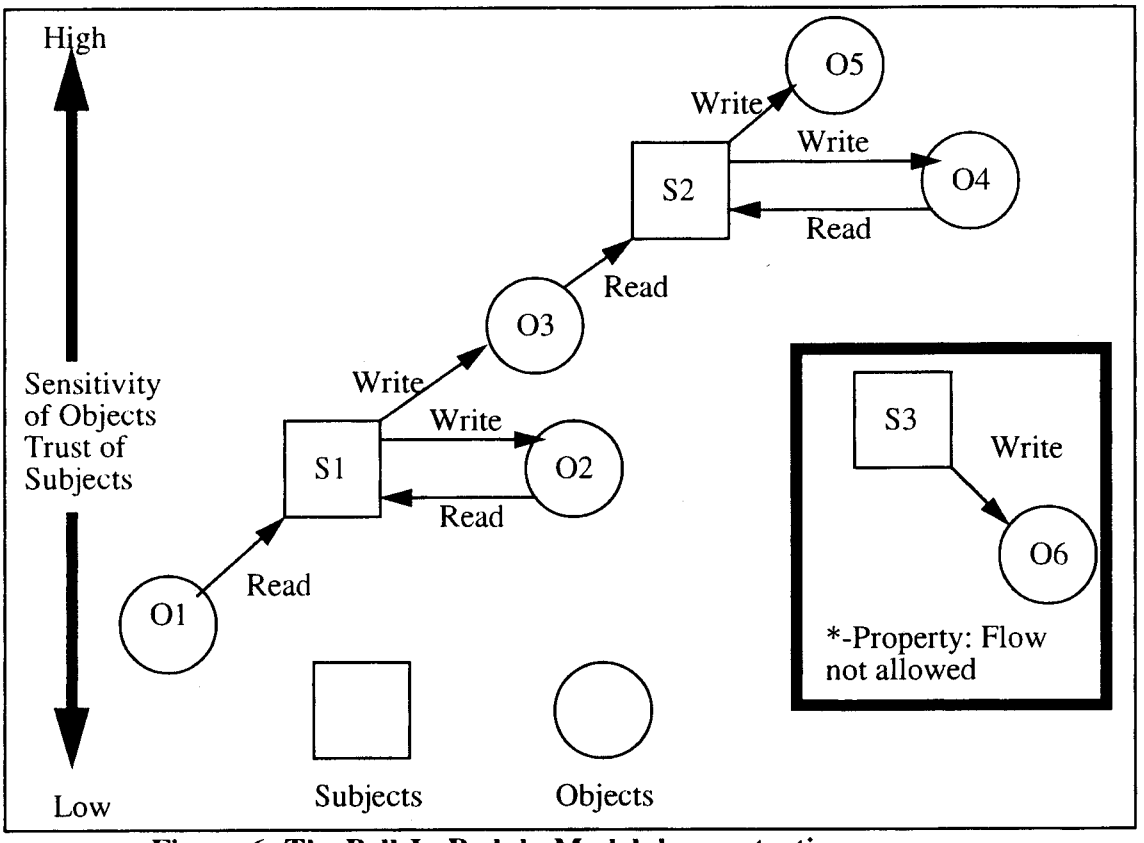

Figure 6: The Bell-LaPadula Model demonstrating the secure flow of information [PFLE89]

The simple security property means that a subject can read objects at its security classification and below. For example, a SECRET subject can read SECRET, CONFIDENTIAL, and UNCLASSIFIED data.

The *-property of the Bell-LaPadula model is used to prevent write-down, so that a subject with a high classification (which has access to high-level information objects) cannot copy information into a lower level object. (See flow demonstrated in Figure 6 in heavy black box; this is not allowed.) This is synonymous with the military security model, which prevents persons with TOP SECRET clearances to give TOP SECRET information away to UNCLASSIFIED users.

Discretionary access control permits the owner of an object, such as a file, to authorize access to that object by other subjects in the system. This creator, at his/her own discretion, determines who is authorized to access the objects he/she creates. Discretionary access control is best demonstrated by way of an access control matrix.

c. *Access Control Matrix*

The access control matrix is a table in which each row represents a subject, each column represents an object, and each entry is the set of access rights for that subject to that object. [PFLE89] This matrix may in fact be sparse because not every row and column intersection will have an entry; most subjects will not have access rights to most objects. In Table 1 below, an access control matrix is shown; objects are shown in the double-boxed columns and subjects represent rows. The allowable modes of operation (i.e., rights) for each subject are: o (owner), r (read), w (write), x (execute). (Note that some boxes are blank).

TABLE 1: ACCESS CONTROL MATRIX

Subject	file a	file b	file c	help.t	compiler	linker	clock	printer
USER A	orw	orw	orw	r	x	x	r	w
USER B	r			r	x	x	r	w
USER C	rw			r	r	x	r	w
USER D			r	r	x	x	r	w
sys mgr	-	-	-	rw	ox	ox	orw	o

Because the access matrix can be represented as a list of triples (subject, object, rights), searching a large number of triples can be time consuming and inefficient. Therefore, the access control matrix is used more as an abstraction than a real implementations.

the rights to declare who has what access and to revoke access by any person whenever desired. Each user has a file directory, which lists all the files to which that user has access.

The UNIX operating system's file ownership and permissions are a good example of this directory type implementation, and as discussed earlier, of discretionary access control. UNIX implements a very simple mechanism which uses only a few bits of access control information attached to each file. Every file has an owner who created that file. The files created by the owner are so listed with that ownership, and file permissions, or modes, are associated with that particular file. A total of ten bits are used to indicate which permissions are applied to three different entities: user (owner), group, and others. Each entity can have permission to read (r), write (w), or execute (e) the file as a program. See Figure 7 below.

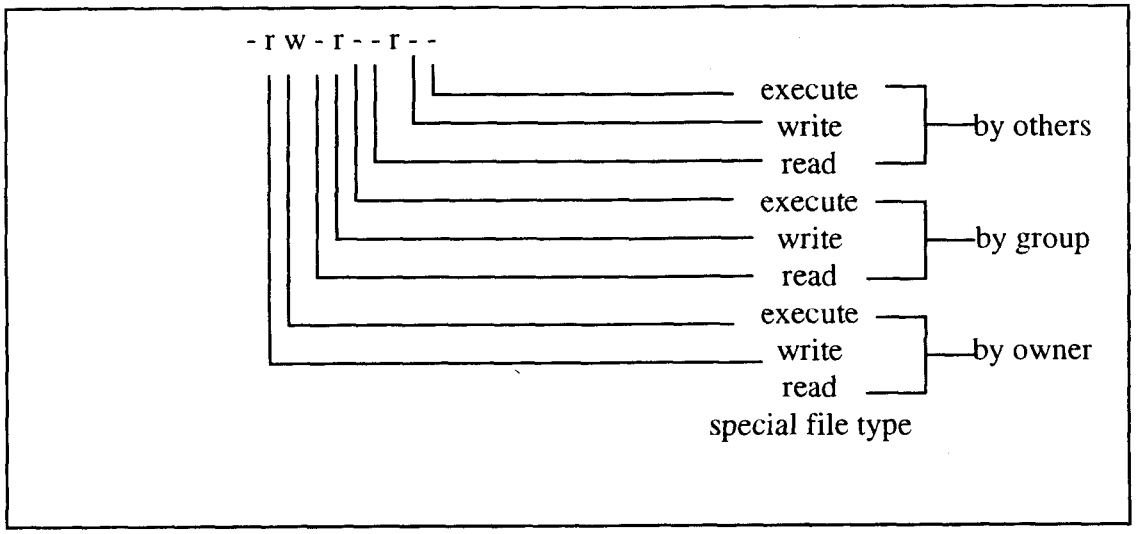

Figure 7: UNIX Protection System

A dash indicates that a permission is not enabled. The left most bit is used to indicate a special file type, like 'd' for directory. Typical files will have a dash in this location.

2. Access Control List System

Another access control mechanism and one of the most effective access control schemes is the access control list (ACL) [GASS88]. In this implementation every object

H. TRUSTED COMPUTING BASE

We have now traversed to the point where we must now place within the system the security mechanisms we have implemented (i.e., protected directories, ACLs, capabilities, etc.). The location where we placed the security mechanisms is inside a perimeter we call the trusted computing base. The trusted computing base (TCB) has been defined by the TCSEC as:

> the totality of protection mechanisms within a computer system, including hardware, firmware, and software, the combination of which is responsible for enforcing a security policy. "A TCB consists of one or more components that together enforce a unified security policy over a product or system. [DOD85]

The TCB contains all the necessary mechanisms needed to provide for the security of the computing system in accordance with the defined security policy. The incorrect operation of the mechanisms within the TCB could lead to the unauthorized disclosure of information or another security violation relative to the system's security policy.

To further define and develop this notion of the TCB, the boundaries of the system must be identified. Two boundaries are of importance as discussed in [GASS88] are the system boundary and the security boundary.

1. System Boundary

The computer system's boundary or interface with the outside world, must be clearly defined, and the threats from the outside world must be identified and evaluated before a security policy can be developed. The system is composed of all the computing hardware, firmware, and software, and includes all the telecommunication hardware and software as well (i.e., networks, phone-lines, wireless, etc.). Everything identified as being inside or part of the system, must be protected by the system. Everything outside the system is left unprotected by the system. [GASS88] The threats to the system must be made a primary focus during the security plan development. [GASS88]

within the security perimeter must be precisely defined, because once they malfunction, a security violation can occur.

This interface, called the security boundary, must be well defined just like the system boundary interface. [GASS88] This interface is controlled and enforced by the system's security relevant components. In a system which utilizes a security kernel, the list of system calls between the security kernel and the operating system is a good example of the interface between the two components of the system. In Figure 9 below, the system's hardware, security kernel, a portion of the OS, and a portion of the DBMS comprise the TCB. (Builders of secure systems try to minimize the size of the TCB to make validation easier.) The users of the system are outside the system and the external controls prevent known threats from entering the system. The TCB is maintained by the security relevant components of the system and is responsible for all security decisions.

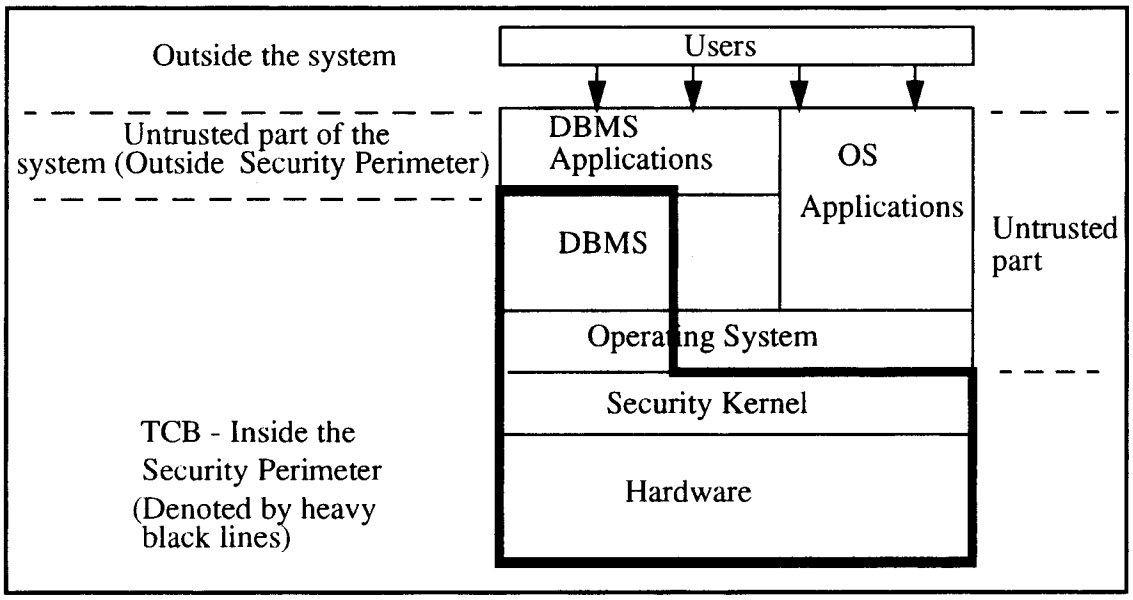

Figure 9: System Boundaries and the Trusted Computing Base

The TCB and the internal security controls, which comprise the TCB, are the primary focus of the design and evaluation of trusted systems.

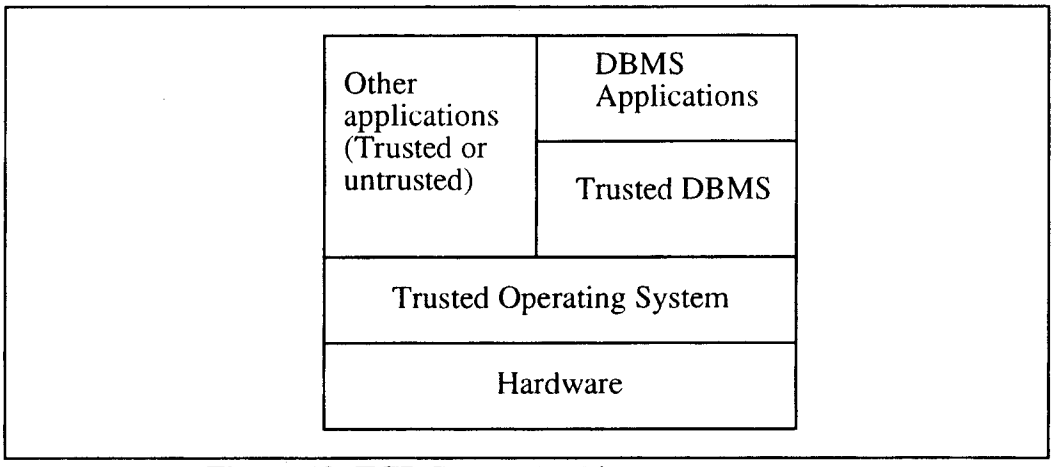

Figure 10: TCB Subset Architecture

An advantage of the TCB subsetting approach is that it allows vendors building an upper level TCB to take advantage of the security features provided from the lower TCB upon which they build their new product. For example, if a trusted operating system provides for mandatory security between subjects and objects, then the newly built DBMS need only enforce additional discretionary security needed by the database. Thus, building multilevel DBMSs using this approach may be the quickest and the most viable approach to getting a multilevel DBMS product evaluated. [LUNT92]

The use of TCB subsetting also can provide the greatest degree of security possible for mandatory security.[LUNT92] Because there is no trusted MAC component in the DBMS itself, the risk of disclosure of sensitive data is considerably reduced. This is because the DBMS is governed by the mandatory TCB of the underlying operating system, which partitions multilevel data by their classification. Thus the subjects within the DBMS, when operating on behalf of the users, cannot gain access to any data whose classification is not dominated by the users' clearance. This means that database operations can be handled by subjects which are single-level and untrusted with respect to mandatory access controls. This is the most conservative approach possible for mandatory security. [LUNT92]

the trusted subject does not permit unauthorized information flow, as opposed to showing that it correctly enforces an access control policy. [NCSC89]

One disadvantage of the trusted subject methodology is the definition of how the trusted subject (i.e., DBMS) is used, because its use can cause new information flows beyond those that can be discovered by performing a flow analysis exclusively of the trusted subject. Such flows can be discovered only by performing a flow analysis on the combination of the trusted subject and the underlying TCB (i.e., the operating system), a task which may in and of itself be very difficult to accomplish.

Over the past decade, as the Criteria have matured (i.e., new interpretations issued) and additional publications (i.e., from the Technical Guidelines Program) have emerged, more commercial developers have moved towards building secure computer systems. In the beginning, manufacturers built systems using the Criteria because it was mandated by the U.S. Government. However, now commercial and private companies have realized the need to better protect proprietary information, personnel databases, and other private or sensitive information. Because of this, the use of trusted systems is becoming more widespread outside the government establishment [GASS88].

B. HISTORY OF THE TCSEC

The National Computer Security Center (NCSC) is part of the National Security Agency (NSA), an agency of the U.S. Department of Defense (DOD). In January 1981, the Department of Defense assigned the responsibility for computer security to the Director of the National Security Agency (NSA). This action led to the formation of the Computer Security Center, whose charter was promulgated in the DOD Directive 5215.1 in October 1982. It specifically tasked the Computer Security Center to establish and maintain:

> technical standards and criteria for the security evaluation of trusted computer systems that can be incorporated readily into the Department of Defense component life-cycle management process.[NCSC90]

The NCSC, in conjunction with other components of the NSA (e.g., Information Systems Security Organization-ISSO), is involved in establishing computer security criteria and guidelines such as the TCSEC, evaluating computer hardware and software products for security and assurance against the Criteria, and conducting and supporting computer security research and development.

Before the Computer Security Center was established, two departments of the U.S. government were instrumental in the establishment of computer security standards and criteria for evaluating computer system products. They were the DOD and the Department of Commerce (DOC).

C. THE CRITERIA

The concept of the trusted computing base (TCB) is fundamental to the understanding of the TCSEC. (See "TRUSTED COMPUTING BASE" on page 29.) Once the TCB can be identified, evaluated, and rated, a level of assurance (rating class) can be given to the product and the system can be considered a trusted system. The Criteria contains three basic control objectives: security policy, accountability, and assurance.

1. Security Policy

The security policy of an organization is the starting point for any implementation of external and internal security mechanisms, and is a basic control objective of the TCSEC. The security policy must be defined in terms of the perceived threats, risks, and goals of an organization [DOD85]. The people or users of the system must be identified, and all the information that will be stored in the system must be located and distinguished from non-system information.

2. Accountability

Another control objective of the TCSEC is the accountability of subjects, which includes I&A and audit capabilities [DOD85]. Each access to a trusted system by a user must be mediated by a security control mechanism which correctly identifies individual subjects (by authenticating a password or other indelible unique feature) and controls what classes of information that subject can access to. A record of all security relevant actions (audit record) by the users must be kept so that any responsible party can be traced after a system violation has occurred.

3. Assurance

Assurance is the guaranteeing or providing of confidence that the security policy has been enforced correctly; that the reference validation mechanism does in fact do its job accurately by implementing the intent of the security policy. The security mechanisms which enforce the security policy must be capable of being "independently evaluated" so

(high assurance) are: D, C, B, and A. Overall there are seven different ratings (classes) that a product or system can earn; D,C1,C2,B1,B2,B3,A1.

1. Division D (Minimal Protection)

Division D contains one class (Class D), and is reserved for all computer systems or products that have failed to adequately meet the requirements of another higher evaluation class. Class D products or systems cannot be expected to protect any security policy or even human error.

2. Division C (Discretionary Protection)

Division C contains Class C1 and Class C2. These classes provide some confidence that the TCB is enforcing a discretionary security policy. The particular items of interest are discretionary protection, audit capabilities and verification and testing.

Tests must be conducted at Class C1 which verify that the security mechanisms (DAC, I&A) work in accordance with the system documentation. A level of assurance must be present that a user cannot by-pass or defeat the security mechanisms of the TCB. Additionally at Class C2, evidence must be demonstrated that I&A data and the audit data cannot be manipulated or destroyed by an unauthorized user. A search for obvious flaws must be conducted, so that any violations of resource isolation or unauthorized access to audit or authentication data is found. There must be "hands-on" involvement in the conduct of independent tests run by the evaluation team. [DOD85]

3. Division B (Mandatory Protection)

Division B contains three classes (B1,B2,B3) and introduces several new design requirements. Most significantly, this division introduces mandatory access controls, labeling of objects and subjects, covert channel analysis, and the requirement that the reference monitor concept be utilized in the Class B2 and B3.[DOD85]

It is at the Class B2 that serious security concerns are realized. At the lower classes of assurance, security can be thought of after-the-fact; an already designed system

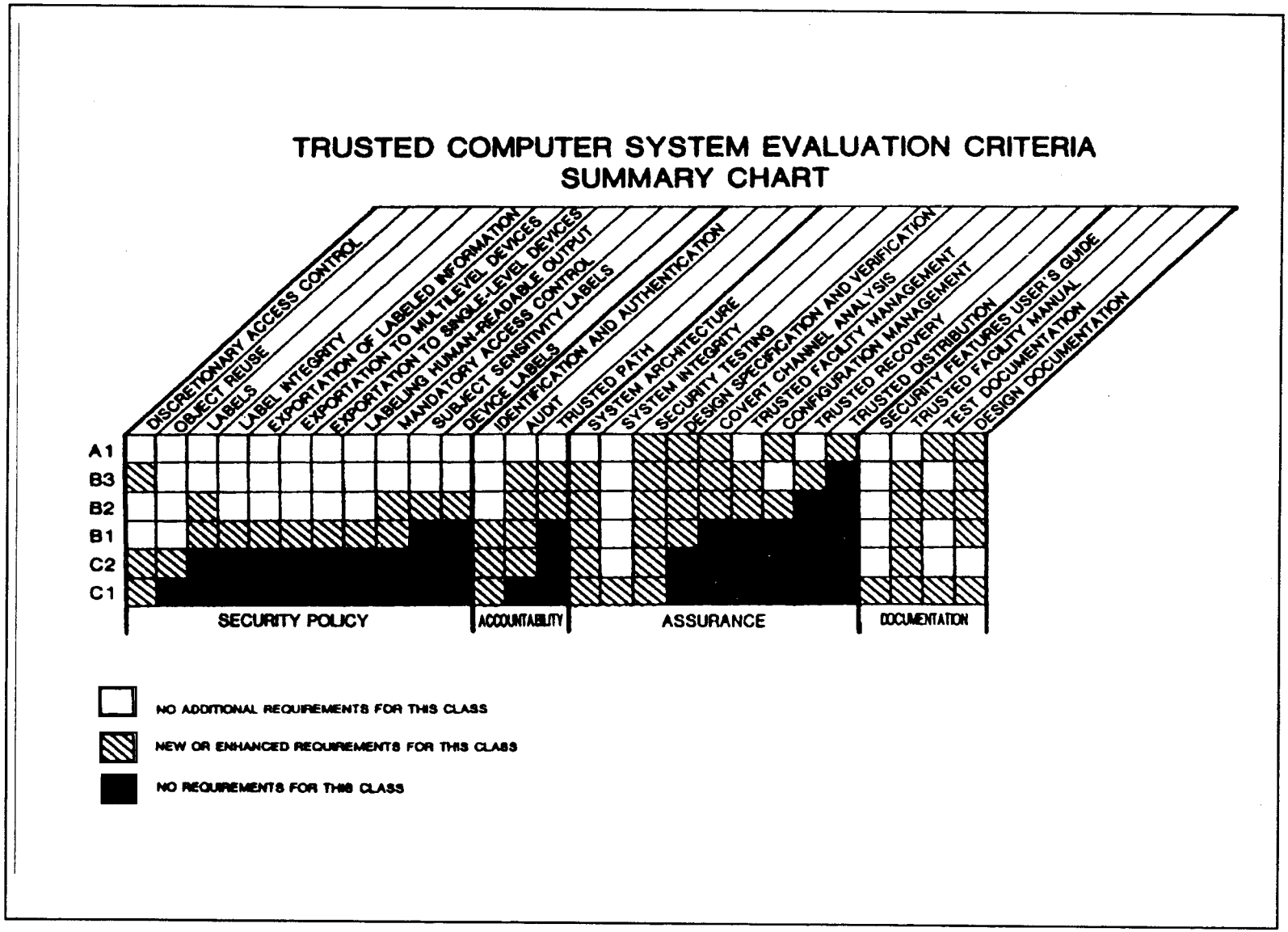

Figure 11: TCSEC Summary Chart from

White-Red Book," Great Britain has the "Green Book" and Canada has the Canadian Trusted Computer Product Evaluation Criteria (CTCPEC). [TROY92]Since the birth of the European community as a political and economic entity, a more coordinated approach of defining computer security standards was needed. Four European countries (Germany, France, Great Britain, and the Netherlands) combined their resources to create the *Information Technology Security Evaluation Criteria* (ITSEC).[TROY92]

1. European ITSEC

Because the European community wanted to maintain commonality with the U.S., the members chose the TCSEC as a basis and elected to expand it, adding additional criteria and more detail. [TROY92] Version 1 of the ITSEC was published in June of 1990, and the second version released June 28, 1991. A number of evaluations have already been conducted against the ITSEC, including DBMS evaluations in the United Kingdom.

2. Canadian TCPEC

The Canadian Computer Security Establishment (CCSE) published the *Canadian Trusted Computer Product Evaluation Criteria* (CTCPEC) with influences from the Orange Book and the ITSEC.

3. Federal Criteria/Common Criteria

During the early 1990's, there began a move to consolidate a common federal criteria that would be more in-line with the proposed European ITSEC. The original goal of the Federal Criteria project was to create a U.S. national standard for computer and information system security that according to [CAMP94] would:

- protect previous investment in trust technology
- add value to current criteria
- develop a framework for defining new customer requirements
- promote international harmonization of criteria

This standard was intended to provide information on how to specify requirements for Information Technology product security, to include a fundamental

IV. HP-UX BLS OPERATING SYSTEM

A. BACKGROUND INFORMATION

It is mandatory that we look at the operating system in our security analysis of both Trusted ORACLE and Informix On-Line/Secure. The reason is that the operating system is an extremely important subset of the trusted computing base. We have purposely chosen a common operating system, HP-UX BLS Version 8.0, so as to both limit the scope of the thesis (i.e., now only one operating system must be partially examined, instead of two), and to make our comparison of products have a common foundation (i.e., a common OS).

1. History

The UNIX operating system was developed by Ken Thompson of AT&T's Bell Labs in the late 1960's as a general purpose interactive timesharing system. After further refinements by the researchers at Bell Labs, UNIX became widely available in 1975. The University of California at Berkeley led the way in making many improvements to the system and began releasing their own improved versions called BSD (Berkeley Software Distribution). Meanwhile, AT&T continued to make improvements to their original system (System V) and thus released many new versions in the years to follow. These two versions, Berkeley's BSD and AT&T's System V, were in widespread use by the mid 1980's.

The standard Hewlett-Packard Unix (HP-UX) is based on and is compatible with UNIX System Laboratories (USL's) UNIX operating system [EDGE93]. USL's UNIX is similar to the Fourth Berkeley Software Distribution Unix software. Therefore, it has many of the characteristics of the Berkeley Unix operating system. However, HP-UX B-Level Security (BLS) does not support all the functionality of its predecessor, HP-UX. HP-UX BLS is a security enhanced version of HP-UX designed to meet the requirements of a Class B1 system [HEWL92a].

TCB includes the processor and I/O internal buses, bus adaptor cards, disk drives, tape drives, and printers.

2. How UNIX Works

This is a brief overview of how UNIX works as taken from [TANE92].

> A UNIX system, can be regarded as a kind of pyramid. At the bottom is the hardware, consisting of the CPU, memory, disks, terminals, and other devices. Running on the bare hardware is the UNIX operating system. Its primary function is to control the hardware and provide a system call interface to all application programs. These system calls allow user applications to create and manage processes, files, and other resources.[TANE92]
>
> Application programs make system calls by putting arguments in registers and issuing trap instructions to switch from user mode to kernel mode to start up UNIX. A library is provided, with one procedure per system call. Each procedure first puts its arguments in the proper place, then executes the trap instruction. The trap instruction performs the required task and then returns to the user mode, where the application program is started again. [TANE92]

The operating system is a resource manager. It performs primitive functions to assist the application programs by controlling such things as the processors, memory space, and I/O devices.

C. SECURITY ENHANCEMENTS

As previously noted, HP-UX BLS is a security-enhanced version of the standard HP-UX operating system. A number of changes have been implemented to meet the Class B1 evaluation requirements. In the following sections, we will give a brief description of some of the security features that are implemented in the HP-UX BLS system.

1. Administrative Roles

One of the significant changes between standard HP-UX and HP-UX BLS is in the area of system administration [HEWL92a]. The system administration tasks have been split into a number of logical roles, thereby enforcing the concept of separation of privilege. The roles are split into functional areas, thus all of the roles can be given to one individual or they can be divided up between different individuals (depending on the needs and the

TABLE 3: PROTECTED SUBSYSTEMS FROM [HEWL92A]

Subsystem	Authorization	Function
Authentication	auth	Assigns authorization and clearances to users
Audit	audit	Maintains and analyzes output from the system's auditing functions
System Administrator	sysadmin	Configures new versions of the operating system and tunes the system
Memory	mem	Allows processes to read memory occupied by the operating system
Backup	backup	Maintains and backs up the file system
Cron	cron	Handles the scheduling of jobs on a delayed or periodic basis
Terminal	terminal	Controls terminal resources of the system
Line Printer	lp	Controls the printer resources of the system. Prints job requests made by users
Tape	tape	Controls the data import/export resources of the system

3. Login/Logout

The HP-UX BLS operating system requires the user requesting access to the system to enter his/her login name, password, and sensitivity level. This sensitivity label must be equal to, or lower than the user's clearance (the highest level the user is cleared for). The system then replies with the user's sensitivity level and the data (i.e, terminal ID) and time of last successful and unsuccessful login attempts.

The system administrator may select to permit user-defined passwords or may require the use of a random password generator. In addition, a password aging function can be selected which will prompt the user when it is time to change his/her password. If the password expires, then the user's account will be locked and the system administrator will be required to re-enable the account [HEWL92c].

The authorization set, although associated with a user, is stamped on all the user's processes. The above sets restrict users and programs in the use of system calls, and they create a mechanism which can be used to implement a policy of least privilege [HEWL92a].

5. Protecting Files

Protecting files in HP-UX BLS is similar to the protections found in standard UNIX, that is, the use of protection bits. (See "Protected Groups or Directories" on page 26.) In addition, because it is necessary to restrict file access to the granularity of a single user (which is not found in standard UNIX) to meet the Class B1 assurance level, access control lists (ACLs) are utilized in HP-UX BLS.

The ACLs are structured to provide three entries: user, group, and protection specification [HEWL92c]. The use of a special character "*", called a wildcard, enables general access to any user meeting the other requirements. Also, the protection specification can use r (read), w (write), x (execute), **all** (for r, w, e), or **null**, **none**, or "---" for no read, write, or execute. See Table 5 below for examples of how ACLs are used in HP-UX BLS.

TABLE 5: ACL ENTRIES IN HP-UX BLS FROM [HEWL92C]

ACL Entry	Explanation
<john.acct,r>	John, when in group **acct**, has read access
<*.acct, r-->	Any user, when in group **acct**, has read access but is denied write or execute
gary.*,null>	Gary, in any group, has no access permission
<*.progs,->	Any user in **progs** group is denied read access
<*.*,--->	Any user in any group is denied access

composed of zero or more nonhierarchical categories, which might include, (in military context) NATO, CRYPTO, and NUCLEAR. [HEWL92a]

7. Import/Export

HP-UX BLS controls all data imported into and from the system. The import medium is the magnetic tape or the floppy diskette; the export medium is the magnetic tape, floppy diskette, or printout. All import or export media are labeled or unlabeled. Labeled media are labeled with a sensitivity label that is recognized by the system (i.e., the system is set up to accept the label). The unlabeled media usually have external stick-on labels (or banner pages for a printout) that represent the sensitivity of the information on the medium.

All import/export devices are designated as either single-level or multilevel devices. Single-level devices are associated with a single sensitivity level and all data imported into or exported from the system is handled at that level. Multilevel devices can determine the sensitivity labels associated with objects imported to or exported from the system and then make the appropriate decisions to place the objects (i.e., files) in the correct directory or disk drive. (In the case of printout, it will place the correct label on each page of the printout by reading the file's label and then printing it on the respective page.) All device information is placed in a security database for retrieval by the access control mechanism.

8. Security Databases

HP-UX BLS requires the maintenance of several security databases to enforce the mandated security policy. (See Table 6 below). The protected subsystems access these databases when needed to obtain information for determining access control. The system administrator can change the parameters within the databases to suit the appropriate security policy.

All the security databases in Table 6 are self explanatory except for the System Defaults database. This database stores default values for the Protected Password, Terminal Control, and File Control databases. If the system administrator does not modify these three

a. Memory Space Allocation

Input and output of both HP-UX BLS and Trusted ORACLE are done in units of storage called blocks. The size of Oracle blocks must be set by the DBA to enhance performance of the DBMS. It is recommended that a block size of 2K bytes be utilized upon initial installation. (The maximum Oracle block size is 8K).[ORAC92b]

b. Database and Log Files Size

The recommended database file size in Trusted ORACLE is 5 MB. A minimum of two log files is required per database and 100K bytes of storage is recommended for each file.[ORAC92b]

c. Filename Restrictions

Trusted ORACLE limits the length of some filenames to a maximum of 14 characters. [ORAC92b] This applies to HP-UX BLS file system when it does not have long filenames enabled within the system.

d. Terminal Characteristics

Some Trusted ORACLE utility programs (i.e, SQL*Plus) use special characters to call files which do not coincide with the characters found in HP-UX BLS. For example, the "@" character in Oracle calls an indirect command file; whereas this same character in HP-UX BLS is the line kill character default. This character should be redefined in Trusted ORACLE to avoid unexpected results.[ORAC92b]

2. Informix Support

Informix On-Line/Secure works with several secure operating systems, each of which has a slightly different implementation. Informix treats specific operating systems as if they are members of the following families: System V MLS, OSF MLS, CMW, and System V, version 4 ES. [INFO93b] HP-UX BLS belongs to the Compartmented Mode Workstation (CMW) family.

V. TRUSTED ORACLE ARCHITECTURE

This chapter and the one following explain the configuration of the DBMSs and the HP-UX BLS operating system. Both systems can be configured in different ways, depending upon the choices made by the persons installing the system.

A. BACKGROUND

Trusted ORACLE 7 is a Class B1 security enhanced package based on the standard Oracle Relational Database Management System, Release 7.0. Therefore, the Trusted ORACLE 7 package includes all the features (functionality) of standard Oracle 7, along with multilevel security.[DATA94b]

1. History

Trusted ORACLE 7 is a distributed server database, first released in 1992. It includes all the functionality found in the standard Oracle 6 (its predecessor), plus a number of new features including a multi-threaded server, query optimizer, row-level locking, and role-based security. [DATA94b]

2. Platforms Supported

Standard ORACLE 7 can be installed on more than 88 different computing platforms [DATA94b]. However, Trusted ORACLE 7 is supported on only the DEC SEVMS and Hewlett-Packard's HP-UX BLS operating systems. Ultimately, the DBMS will be ported to a wide range of secure UNIX and proprietary platforms, including Compartmented Mode Workstations, as they become available from hardware and operating system vendors [EHRS91].

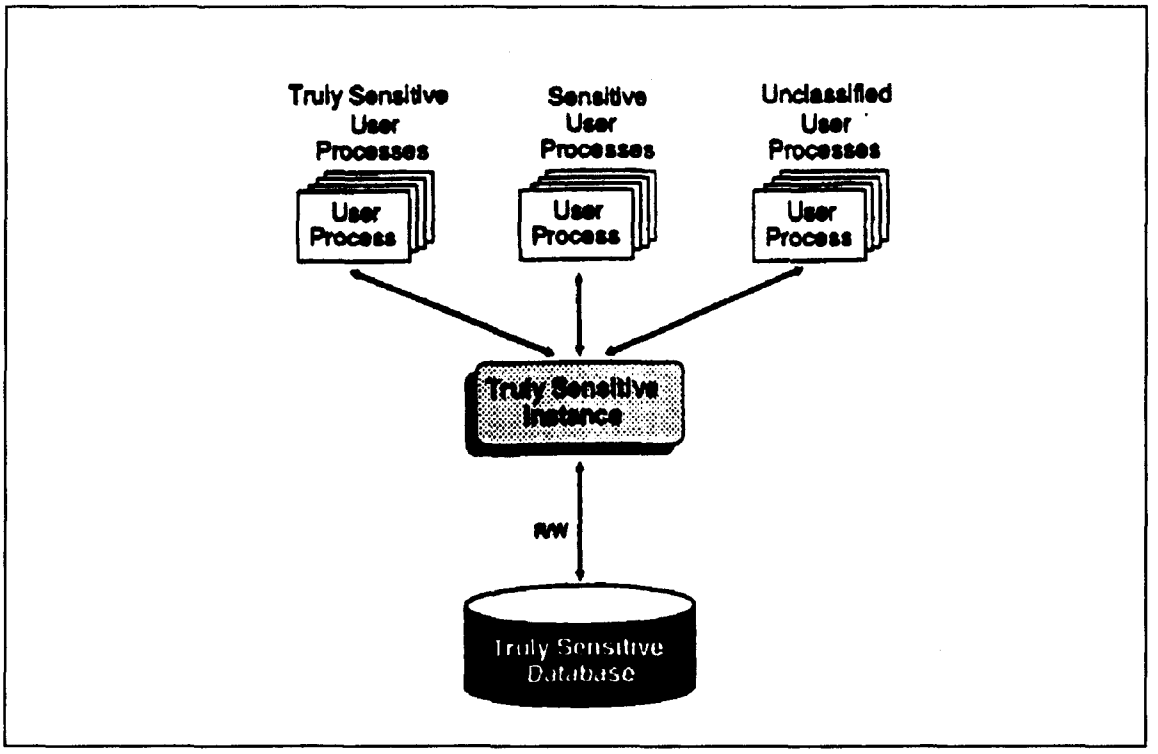

Figure 11: DBMS MAC Mode Database from [ORAC92a]

The DBMS MAC mode will be the mode that we investigate and analyze since it represents a multilevel database in which mandatory access control policy is enforced by the DBMS.

2. OS Mac Mode

In OS MAC mode, multiple, distinct, single-level databases are created, one for each sensitivity label. All mandatory access is mediated by the operating system on the operating system objects (i.e., files) in accordance with the overall security policy. Trusted ORACLE completely relies on the operating system to control access by Trusted ORACLE users to Trusted ORACLE objects. Multilevel tables can be created in OS MAC mode even though a single, physical table cannot contain rows of more than one label. A logical "multilevel" table can by created by identically named tables at each sensitivity label, each with identical attributes. Figure 12 below, demonstrates how a multilevel database system

1. Physical Storage Structures

The following sections briefly describe the physical structures of an Oracle database.

a. Disk Organization

Trusted ORACLE can be set up to utilize raw disk devices. A raw disk device, or raw disk partition, is a hardware device that is supported by a character device driver. A character device driver accesses the raw device through special files that are in the */dev/rdsk* directory. These devices are not buffered by the HP-UX BLS kernel; data is transferred directly between the user's buffers and the device. Raw devices allow I/O directly between the disk where the data is stored and the System Global Area of the Trusted ORACLE server. The overhead of the HP-UX BLS read ahead and file system is avoided, thus performance is enhanced because data is stored together on the raw device.

b. Files

The data files are the files which contain the actual database data. Database schema objects (i.e., tables, clusters, indexes) are physically stored in the data files allocated to the database. A data file cannot change in size once created, therefore as a database grows in size, new data files are added to accommodate the database.

There are two or more redo files for every Oracle database. This set of redo files is known collectively as the "redo log." The redo log's primary purpose is to record all changes to the database. The information in the redo files is used only to recover the database from a system failure when the data has not been written to the data files.

One control file exists for every Oracle database. Its primary purpose is to record the physical structure of the database, such as the database name, the names and locations of the database's data files and redo files, and the time stamp of database creation.

functions such as reads and writes to the database files (i.e., DBWR-Database Writer, and LGWR-Log Writer) and other needed checks and locks.

We have not ascertained from available documentation, what sensitivity level the daemons processes run at, or even if they have a sensitivity label at all. Other information (other than the *Trusted ORACLE User's Guide* and technical overviews) would have to be obtained to find the answers to this question.

d. Blocks, Extents, and Segments

The operating system file system has a specific number of bytes which make up an operating system block. In HP-UX, the block size is usually 2K bytes (2048 bytes). The Oracle database also recognizes, at its highest granularity level of storage, a data block (or page). Oracle allocates all its database space in blocks. This database block can be equal to the operating system block, or a multiple of it (e.g., a database block could be 2K or 4K bytes).

At the next level of storage is the "extent." An extent is a specific number of contiguous data blocks that are allocated for storing a specific type of information [ORAC92c]. For example, if more space is needed to store Oracle data files, then a data extent will be allocated for that data. If more space is needed for control information, then a control extent will be allocated.

The highest level of logical database storage is the segment. A segment is a set of extents which stores a specific type of data structure, such as a database table's data. The relationship between these three database spaces is shown in 14, below.

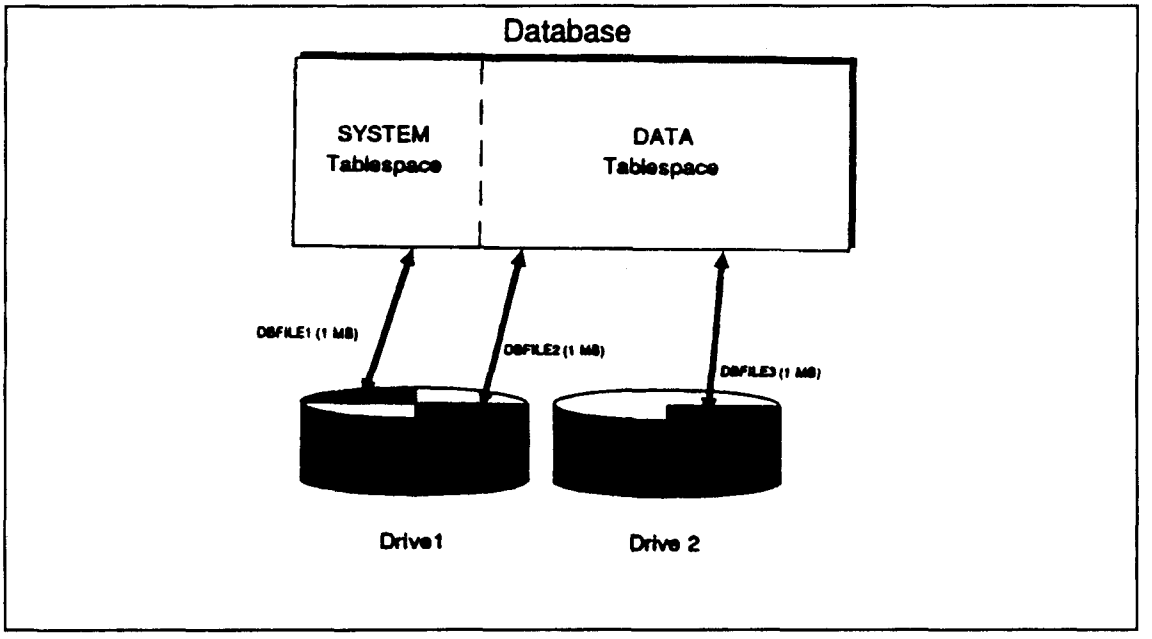

Figure 15: Logical Structures from [ORAC92c]

b. Schema Objects

Most schema objects such as tables, clusters, and indexes are stored within a tablespace. The data for a table is stored in one or more of the tablespace's data files. A cluster is an optional way of storing table data; it groups the tables that share the same data blocks together. This is because some tables share the same columns of identical data and are often used together. Clusters are used primarily to reduce I/O and to reduce the amount of storage space needed by storing redundant data only once.

A view is a tailor-made presentation of one or more tables and it is not stored within the tablespace or any other storage space. The only thing stored in a view is the view query or definition. When a view is invoked, it dynamically queries the appropriate tables stored in the database and then presents the data queried in a table-like format. (A view is often called a "virtual table.")

which are used to make labels human-readable. An example of a numeric might be 100:1, where 100 is the sensitivity level and 1 is the category. A short character format might be TS:A and a long format might be TOP SECRET:NATO. (More will be said about the Informix MAC features in the subsequent chapters.)

2. Privileges and Roles

A privilege in Oracle is a right to execute a particular type of SQL statement. Oracle divides privileges into distinct categories: system privileges, object privileges and MAC privileges (in Trusted ORACLE 7 DBMS MAC mode only.) System privileges allow users to perform particular system-wide functions, such as connecting to the database. Object privileges allow users to perform a specific action on a specific object, such as delete a row on the EMPLOYEE table. MAC privileges allow users to perform operations that circumvent MAC policy, such as reading higher level data.

Each MAC privilege corresponds to a similar privilege in the underlying operating system. A user with granted MAC privileges in Trusted ORACLE cannot execute the command successfully unless the corresponding privilege has been granted in the operating system[ORAC92a].

The three MAC privileges in Trusted ORACLE 7 DBMS MAC mode along with the needed HP-UX BLS privileges are shown in Table 7, below.

TABLE 7: MAC PRIVILEGES IN DBMS MAC MODE [ORAC92B]

MAC Privilege	HP-UX Privilege	Function
WRITEDOWN	downgrade or allowmacaccess	Allows users to perform write operations on data at a lower label.
WRITEUP	writeupclearance, writeupsyshi, or allowmacaccess	Allows users to perform write operations on data at a higher label.
READUP	allowmacaccess	Allows a user to perform read operations on data at a higher label.

VI. INFORMIX-ONLINE/SECURE ARCHITECTURE

This chapter explains the configuration of the Informix DBMS. This chapter is our effort to explain the Informix DBMS structure so as to better prepare the reader for the subsequent comparative analysis of MAC policy enforcement.

A. BACKGROUND

Informix-OnLine/Secure is a multilevel secure relational database management system (RDBMS) for secure UNIX and compartmented mode workstation (CMW) platforms. OnLine/Secure comes in two different versions: B1 and C2. For the purposes of our comparative analysis, we only analyzed the B1 configuration.

1. History

Informix Software, Inc., is a subsidiary of Informix Corp., with corporate headquarters in Menlo Park, California. In September 1993, Informix's OnLine/Secure became the first database to meet Class B1 and Class C2 security levels, as specified by the NCSC (even though Final Evaluation Reports have yet to be made public as of this writing.)

2. Platforms Supported

Informix On-Line/Secure is available for Hewlett-Packard HP 9000 secure system, Sun Microsystems' Sun CMW secure system, SCO, Digital Equipment, Sun SPARC, and Zenith [DATA94b].

B. CONCEPT OF OPERATIONS

Informix-OnLine/Secure operates on a client/server model, where the client front end operates as a separate process from the server's backend process. These two processes

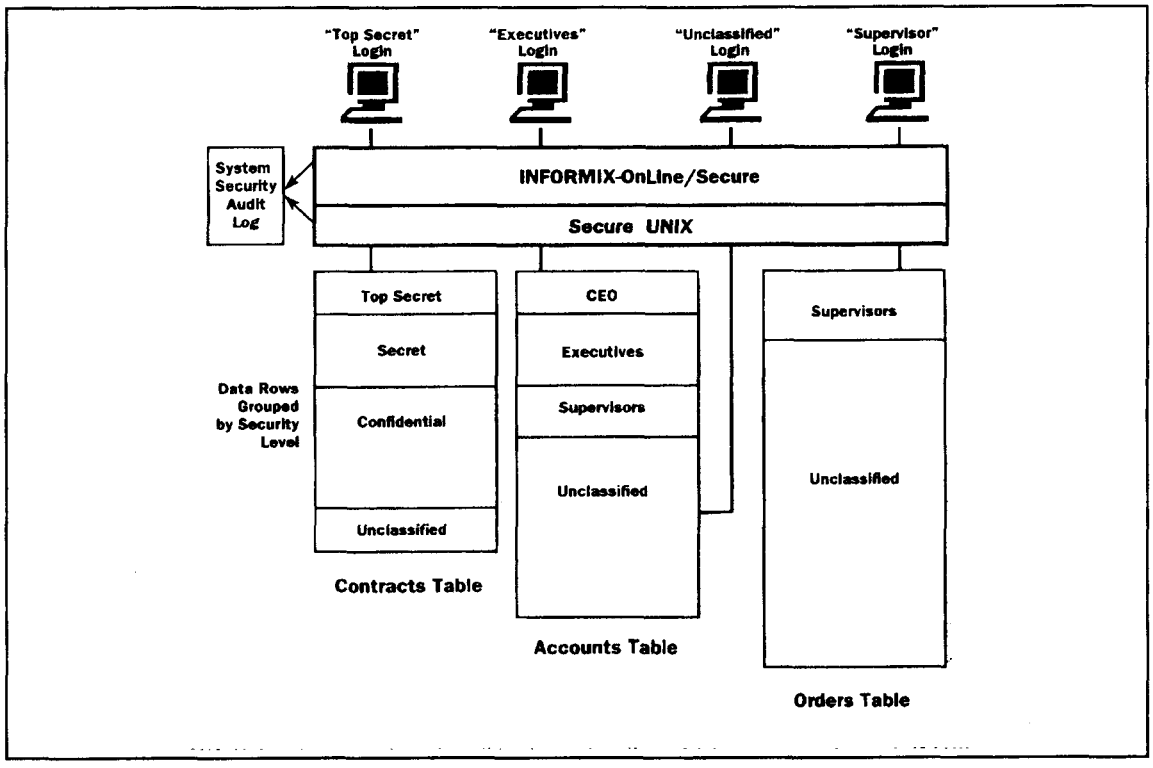

Figure 16: Basic Architecture from [INFO93a]

C. DATABASE STRUCTURES

Informix OnLine/Secure has additional structures and features relative to the standard Informix-OnLine RDBMS server. These new system capabilities are designed to meet the Class B1 assurance level of the TCSEC [INFO93c].

1. Physical Storage Structures

The following sections briefly describe the physical structures of the Informix-OnLine/Secure database.

a. Disk Organization

The Informix-OnLine/Secure database server is designed to perform its own disk management [INFO93d]. Raw devices are identified, (by using a UNIX utility), to be used in the storage of all database data and system catalogs. Raw devices are usually

Shared memory is advantageous for several reasons, including the elimination of buffers for every process, (all database buffers are pooled), thus reducing disk I/O. Buffers are not reread, because only the most recent data page is in memory, and concurrency is enhanced because data is already in memory.

Logical logs record all the changes to the database since the last backup of data was made. The logical log buffers within the shared memory area are used to temporarily hold data before it is written to the logical log disks. The physical log buffers hold a copy of a database page on the disk before the page is changed. These "before-images" allow the system to reconstruct the state of the disk at the time of the last checkpoint (i.e., points in time when the database server knows all databases are consistent) before the system failure occurred.

Disk mirroring is the process of creating a mirror image of data in the database. This mirroring process requires the use of a primary database disk and a mirror disk. Database mirrors are optional in Informix-OnLine/Secure and are utilized for high availability.

c. Chunks, Pages, and Extents

The basic unit of storage in On-Line/Secure is the "chunk." A chunk is a unit of disk storage that has been dedicated to the Informix RDBMS server. Chunks can be either raw devices, parts of raw devices (i.e., partitions), or files under the UNIX operating system.

The page is the basic unit of disk I/O in the Informix database server. All space in every chunk is divided into pages and I/O is done in units of whole pages. The size of the page is the same in all chunks used for tables and is set when the DBMS is installed.

Informix OnLine/Secure allocates disk space on the raw devices in units called "extents." Each extent is a block of physically contiguous pages from the space designated to contain the database. When database users add rows to a table, or new tables,

only pages at that level. If the table contained rows with three different sensitivity levels, there would be one bundlespace and three tblspaces for that table.[INFO93a] (See Figure 18 below.)

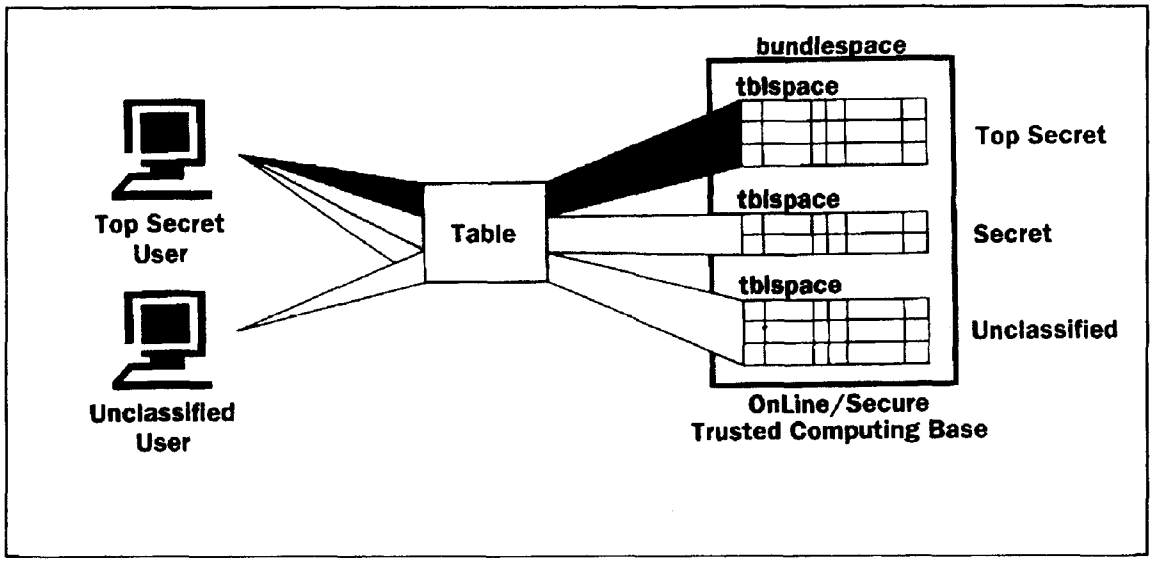

Figure 18: Tblspaces and bundlespaces [INFO93a]

In addition to data pages (which represent the data held in the rows of the table), the tblspaces also contain pages for indexes. Binary large object (blob) column pages are also found in the tblspace, even though the actual blob data is contained in a blobspace (which is similar to a tblspace, except it holds only special blob data types, see below.) [INFO93d]

Blobs are data storage objects that have no maximum size, except for the limitations of the computer, (usually 2^{31} bytes). Blob data types in Informix-OnLine/ Secure are TEXT and BYTE. The TEXT data type is used for storing ASCII data, and the BYTE data type is used for any type of binary data.[INFO93b]

d. Schema Objects

Schema objects in Informix-OnLine/Secure include databases, tables, rows, blobs, views, synonyms, indexes, contraints, and stored procedures. As previously

Sensitivity labels in Informix-OnLine/Secure are represented in four different formats: external, canonical (for System V MLS operating system only), internal, and tag. The external format is a human-readable label such as TOP SECRET:NATO; the internal format is a binary representation. (The canonical format is not used in the HP-UX BLS operating system.) The tag format is a 32 bit integer and is used extensively in the many operations performed on labels, such as label equality and label dominance. A tag is mapped to a human-readable label before exporting the sensitivity label to an output device such as a terminal or line printer, or it may be retrieved by a database user in the tag format.

2. Privileges

Privileges are used in Informix-OnLine/Secure to enforce discretionary access controls. There are three types of DAC privileges in Informix: database privileges, table privileges, and procedure privileges. All privileges are stored in the system catalog tables and any user with the "Connect" database privilege can query the system catalog tables to find out what privileges have been granted and to whom (assuming that this user's session sensitivity level dominates the information in the databases and system tables.) [INFO93c]

Database privileges from lowest to highest are C (Connect privilege), R (Resource privilege), and D (Database Administrator privilege). The Database Administrator privilege is not the same as the Database System Administrator (DBSA) privilege, which is given only to the database administrator. The Database Administrator privilege as mentioned here, allows users to execute the DROP DATABASE (i.e., remove a database from the system) and create DATABASE (i.e., establish a new database in the system) statements.

Eight privileges are applied to tables, which give non-owners the privileges of the owner. Table 8, below, describes each table privilege. A "-" indicates that a user does not possess the privilege; a capital letter, such as "S" allows the user to GRANT the privilege to another user; a small letter "s" does not.

OnLine/Secure MAC or DAC policies. [INFO93c] Discrete privileges allowed in Informix-OnLine/Secure are shown in Table 9.

TABLE 9: DISCRETE PRIVILEGES IN INFORMIX-ONLINE/SECURE

Privilege	Description
PRIV_CANSETLEVEL	Allows the user the ability to alter the session security level at which database operations occur
PRIV_CANSETIDENTITY	Allows the user the ability to alter the user name under which database operations are performed

The PRIV_CANSETLEVEL privilege enables a database user to successfully execute the SET SESSION LEVEL statement, thus effectively changing the session sensitivity level of the user. The PRIV_CANSETIDENTITY privilege enables the SET SESSION AUTHORIZATION statement so that a user can adopt the user name of any non-administrative user. (We will expound on these two privileges later in subsequent chapters.)

3. Auditing

The auditing records produced by Informix-OnLine/Secure events are stored in the operating system audit records only. All use of discrete privileges is audited in Informix-OnLine/Secure as well as all DBSSO actions, initiation of the database system administrator utilities, and each initiation of a new OnLine/Secure session. [INFO93b]

4. Secure Administration Front End

The DBSSO performs most of the security-related maintenance tasks using the secure administrator front end (SAFE). All auditing masks, MAC sensitivity labeling of objects, DAC privilege changes, and granting and revoking discrete privileges are done at the SAFE console. The SAFE provides an interface to the TCB and is part of the TCB. Only the DBSSO is allowed to perform operations at the SAFE.

The next level is basically a repeat of the upper level, except that the TCSEC requirements have been decomposed into more granular requirements. This allows more detailed analysis and a better understanding of what the overall criteria is trying to relay. Likewise, if substantive interpretations existed within the TDI, they to would be decomposed. However, as the present TDI exists, no substantive decomposition could be made.

The lowest level represents the decomposed criteria with individual "line item" interpretations thrown in. These "line item" interpretations are issued from time to time by the NCSC and published in the "Announce forum" of the Dockmaster bulletin board. We have only incorporated "line item" interpretations through September 1993, the time that the first DBMS (Informix-OnLine/Secure) completed substantive evaluation by NCSC. These interpretations are summaries and where found in *INFOSEC Handbook: An Information Systems Security Reference Guide* [ARCA93]. More recent interpretations would have to be accessed via the Dockmaster Announce forum.

Based on this simple methodology our analysis was conducted. Once both products are matched against the decomposed criteria and "line item" interpretations, they will be compared. The comparison of DBMS products is found in Chapter X.

A. TCSEC CRITERIA CHOSEN AND WHY

Informix-OnLine/Secure 5.0 and Trusted Oracle 7.0 have both completed NCSC evaluation for Class B1 - Labeled Protection. The TCSEC Class B1 assurance level was discussed briefly in Chapter III, and will not be further expanded upon here. However, it should be noted, that the Class B1 level of assurance is characterized chiefly by the requirements for labels on some subjects and objects, a suitable MAC policy, and a mandatory access control mechanism implemented to enforce access by these labeled subjects to objects. Other new requirements do exist, such as design specification and verification, and security testing. However, we have characterized Class B1 assurance (labeled protection) as chiefly the implementation of a mandatory access control

1. Key to Understanding Decomposed Statement Notation

The following table (See Table 10, below) constructed from [ARCA93], explains the notation used in the decomposed Class B1 criteria.

TABLE 10: DECOMPOSED CRITERIA NOTATION

Notation	Explanation
{ }	Text in braces replaces original TCSEC text, often done to replace a pronoun with its reference.
[]	Text in brackets is repeated from a previous criterion or is new text included for clarity.
...	Ellipses show were TCSEC text is omitted, typically done when a single TCSEC sentence divides into multiple criteria.
Italics	Italicized text denotes a TCSEC interpretation. Each of these criterion is followed by an interpretation number that generated it.

The TCSEC criteria interpretation summaries (in italics) are included adjacent to the specific criterion they affect. The NSA, over the years, has made a number of criteria interpretations (including discussion of alternate approaches, rationale, and presentation of the selected approach). The TCSEC criteria interpretations are independently numbered, with the assurance class, type interpretation, and the date the interpretation was published. For example, the LAB.i1 is referenced by C1-CI-03-89, which means that this interpretation starts at Class C1, is a criteria interpretation (CI), and was issued by NCSC in March 1989.

2. Labels

Labels are attributes associated with some subjects and objects in a Class B1 multilevel secure DBMS. These attributes represent the sensitivity or classification level of the subjects and the objects. The TCB is required to maintain these attributes for use by the access mediation mechanism.

3. Label Integrity

Label integrity is concerned chiefly with maintaining the correct label on the respective subjects and objects, and ensuring that the TCB protects these labels from modification.

LI.1 - Sensitivity labels shall accurately represent security levels of the specific subjects or objects with which they are associated.

LI.2 - When exported by the TCB, sensitivity labels shall accurately and unambiguously represent the internal labels...

LI.3 - [When exported by the TCB, sensitivity labels]... shall be associated with the information being exported.

TABLE 12: LABEL INTEGRITY SUMMARY

	Trusted Oracle Platform		**Informix Platform**	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS
LI.1	B	B	B	B
LI.2	B	B	D	NA
LI.3	B	B	D	NA

4. Exportation of Labeled Information

From the TCB perspective, the exportation of labeled objects must maintain the integrity of the sensitivity label of the data with the I/O device which receives or transports the data out of the database.

EL.1 - The TCB shall designate each communication channel and I/O device as either single-level or multilevel.

EL.2 - Any change in {the single-level or multilevel} designation {of a communication channel} shall be done manually...

EL.3 - [Any change in {the single-level or multilevel} designation {of a communication channel}] shall be auditable by the TCB.

EM.i1 - *Mulitilevel tape systems are not required to store an object's sensitivity label on the same tape as the object as long as this label can be associated with the object in a trusted manner.* (C1-CI-05-84)

EM.3 - [When the TCB exports an object to a multilevel I/O device, the sensitivity label associated with that object shall also be exported]... in the same form (i.e., machine-readable or human-readable form).

EM.4 - When the TCB exports... an object over a multilevel communication channel, the protocol used on that channel shall provide for the unambiguous pairing between the sensitivity labels and the associated information that is sent...

EM.5 - [When the TCB]... imports [an object over a multilevel communication channel, the protocol used on that channel shall provide for the unambiguous pairing between the sensitivity labels and the associated information that is]... received.

TABLE 14: EXPORTATION TO MULTILEVEL DEVICES SUMMARY

	Trusted Oracle Platform		**Informix Platform**	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS
EM.1	B	B	B	B
EM.2	NM	OS	D	NM
EM.i1	U	U	U	U
EM.3	NM	OS	B	B
EM.4	NM	OS	NM	OS
EM.5	NM	OS	NM	OS

6. Exportation to Single-Level Devices

The TCB must maintain single-level data exported to single-level devices by selecting the output device's sensitivity level based on the information being exported or imported. If data being exported or imported is SECRET, then the device chosen for the input/output should also be SECRET.

HRO.3 - The TCB shall, by default, mark the top and bottom of each page of human-readable, paged, hardcopy output (e.g., line printer output) with human-readable sensitivity labels that properly[2] represent the overall sensitivity of the output or that properly represent the sensitivity of the information on the page.

HRO.4 - The TCB by default and in an appropriate manner, mark other forms of human-readable output (e.g., maps, graphics) with human-readable sensitivity labels that properly represent the sensitivity of the output.

HRO.5 - Any override of {human-readable sensitivity label} marking defaults shall be auditable by the TCB.

TABLE 16: LABELING HUMAN-READABLE OUTPUT SUMMARY

	Trusted Oracle Platform		**Informix Platform**	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS
HRO.1	NM	OS	NM	OS
HRO.2	NM	OS	NM	OS
HRO.3	NM	OS	NM	OS
HRO.4	NA	NA	NA	NA
HRO.5	NM	OS	NM	OS

8. Mandatory Access Control

The mandatory access control requirements address how labeled subjects access labeled objects, and if the access rules, (as stated by the security policy), are enforced by the MAC mechanisms.

MAC.1 - The TCB shall enforce a mandatory access control policy over all subjects... under its control (e.g., processes...).

MAC.2 - The TCB shall enforce a mandatory access control policy over all... storage objects [under its control] (e.g.,... files, segments, devices).

TABLE 17: MANDATORY ACCESS CONTROL SUMMARY

	Trusted Oracle Platform		Informix Platform	
MAC.1	B	B	B	B
MAC.2	B	B	B	B
MAC.3	B	B	B	B
MAC.4	B	B	B	B
MAC.5	B	B	B	B
MAC.6	B	B	B	B
MAC.7	B	B	B	B
MAC.8	NM	OS	NM	OS
MAC.9	B	B	B	B

C. TDI INTERPRETATIONS

The Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria (TDI) was completed and issued in April 1991. One would expect that this publication might contain particular answers to questions related to trusted DBMSs. However, the TDI did not provide us with the revealing answers that we sought.

Section TC-5 of the TDI contains the "General Interpreted Requirements" for DBMS criteria. Often, the TDI added little other than a statement that the requirements of the TCSEC still applied.

For example, we have focused exclusively on Class B1 level Labels and Mandatory Access Control requirements for our evaluation and comparison of products. The general interpreted requirements for labels as stated in the TDI is:

> This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of it's subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

VIII. ORACLE ANALYSIS

This chapter analyzes the Trusted ORACLE 7 against the TCSEC requirements (as decomposed in Chapter VII) for labels and mandatory access controls. We start by looking at the DBMS TCB component (i.e., database server software and user's manuals), then proceed to the operating system TCB component. As discussed in Chapter VII, a requirement, as listed below, can be met in the DBMS TCB component, the operating system TCB component, both components, or it may not be meet or is not applicable. If users are required to meet this requirement, then a "U" will be placed in the respective columns associated with the requirement. After each requirement, we determine where the requirement was met, if in fact they were met. (See requirement summaries in Chapter VII.)

A number of the individual decomposed TCSEC requirements are substantially the same (some are exactly the same). Therefore, we will refer the reader to specified requirements in lieu of discussing the same requirement in two different places.

A. LABELS

The decomposed label requirements are discussed below.

1. LAB.1 Requirement

Subjects are the active processes in the system, be they user subjects or daemon (background) subjects. Daemon subjects are maintained within the respective database server and operating system TCB components; each component has its own set of daemon subjects which are created by the OS. Because we lack the appropriate documentation, we can not discuss daemon subjects specifically, and therefore will not analyze daemon subjects further.

The maintenance of a user subject's label begins with the creation of a username (i.e., account) for the Trusted ORACLE DBMS. All users must have a valid username before they can access the database. When an account is created for a new user on the Trusted ORACLE server, the account definition is stored as a row in a data dictionary table.

maintaining the Protected Password database within the operating system). Because authentication is performed by the OS, the user name and password of the operating system is the one used within Trusted ORACLE to authenticate that the user logging into Oracle is in fact a valid user. For example, if a user with an operating system account named "Ron" is to connect to the Trusted ORACLE database, there must be a corresponding database user "Ron" in the ALL_USERS table within the database data dictionary. When "RON" connects to the database (by typing "/"), the DBMS checks to see if there exists a valid user "Ron"; if so, then "Ron" can begin using the database.

Therefore, both components (the OS and the DBMS) are needed to maintain the user subject levels of the system and a "B" is given to each component in the summary tables in Chapter VII.

2. LAB.2 Requirement

As in the case of user accounts, all object definitions created in Trusted ORACLE are maintained as a row in a data dictionary. The objects found in Trusted ORACLE are database(s), tablespaces, rows, tables, views, indexes, clusters, sequences, synonyms, stored procedures and functions, packages, triggers, and rollback segments. The row for an object definition is labeled at the creator's label when the object is created. The data dictionary is located in the SYSTEM tablespace within the database. This tablespace is made up of segments which correspond to a set number of operating system blocks. Each tablespace is labeled when it is created. All objects placed within a tablespace must dominate the label of the tablespace. You cannot store a lower level object in a higher level tablespace [ORAC92a].

The storage objects seen by the HP-UX BLS operating system (i.e., files, directories, devices, IPC objects, symbolic links, named pipes, processes, printer queues, and ptys) are created and maintained in the OS. Files and directories are labeled individually, and are labeled in such a way that the access classes increase as you go down the tree. For example, the root directory is labeled at System Low and the directories and

5. LAB.4 Requirement

The Trusted ORACLE DBMS utilizes an Import utility to import single level data, from an operating system file into the database. By default, the Import utility performs a single level import on a single level file. (A multilevel OS file can also be imported as single level.)The user importing the data has the responsibility for logging into the system at the level to match the data's label. Therefore, the user is responsible for meeting this requirement in the DBMS component.

The HP-UX BLS operating system handles the importation of Trusted ORACLE database files from outside the system. HP-UX specifically defines two types of import media, labeled and unlabeled. An unlabeled medium is one whose data does not include sensitivity labels. The unlabeled medium typically has some external label (such as a stick-on label for magnetic tape) which tells the user how to handle the data on the medium. This unlabeled medium must then be loaded on a single-level device (associated with a single sensitivity label) which corresponds to the label on the medium. When the data on this medium is loaded into the system, it is labeled at the same sensitivity as that of the single-level device.

Therefore, this requirement that the TCB shall request and receive the security level of the data, is accomplished directly by the authorized users of the system when they properly load tapes or floppy diskettes at the correctly labeled input device. Likewise, we label the HP-UX column of the summary table with a "U", and place an "NA" in the DBMS column since the Trusted ORACLE Import utility is only good if the files to be imported were created by the Oracle export utility.

6. LAB.5 Requirement

There is no mention of auditing the import of non-labeled data in the *Trusted ORACLE Administrator's Guide* [ORAC92a].

The HP-UX BLS operating system allows for the collection of audit data through the use of the Audit System Collection Mask. One of the audit capabilities of this

The Export utility of Trusted ORACLE writes data from an Oracle database into operating system files in the Oracle binary format, either in the format of MLSLABEL or the RAW MLSLABEL [ORAC92a]. Within the Trusted ORACLE DBMS, the MLS keyword tells the Export utility whether or not to export labeling information along with the data a user is exporting. The default is Y (yes), which tells the Export utility to include the ALTER SESSION SET LABEL and the ROWLABEL pseudo-column. (The ALTER SESSION SET LABEL command ensures that when the export file is imported later, the imported objects are recreated at their original labels.) The ROWLABEL pseudo column values contain either of the two MLSLABEL datatypes.

This requirement is met by both the DBMS and OS TCB components, and is so reflected in the summary tables. (See table 12 on page 87)

3. LI.3 Requirement

As stated previously in LI.2, the sensitivity labels are associated with all objects created in the database (e.g., when a database table is created, the ROWLABEL pseudo-column, is automatically created as a special attribute) and tagged at the level of the user process creating the object. Therefore, when a multilevel export is conducted, the Export utility writes information to an operating system file, which includes labeling information for the data exported.

This requirement is met by both the DBMS and OS TCB components, and is so reflected in the summary tables. (See table 12 on page 87)

C. EXPORTATION OF LABELED INFORMATION

1. EL.1 Requirement

The functions to designate each communication channel and I/O device are found within the underlying operating system, HP-UX BLS. The system administrator defines the security characteristics of each import/export device by placing the required information in the Device Assignment database. Every device that is to be used must have

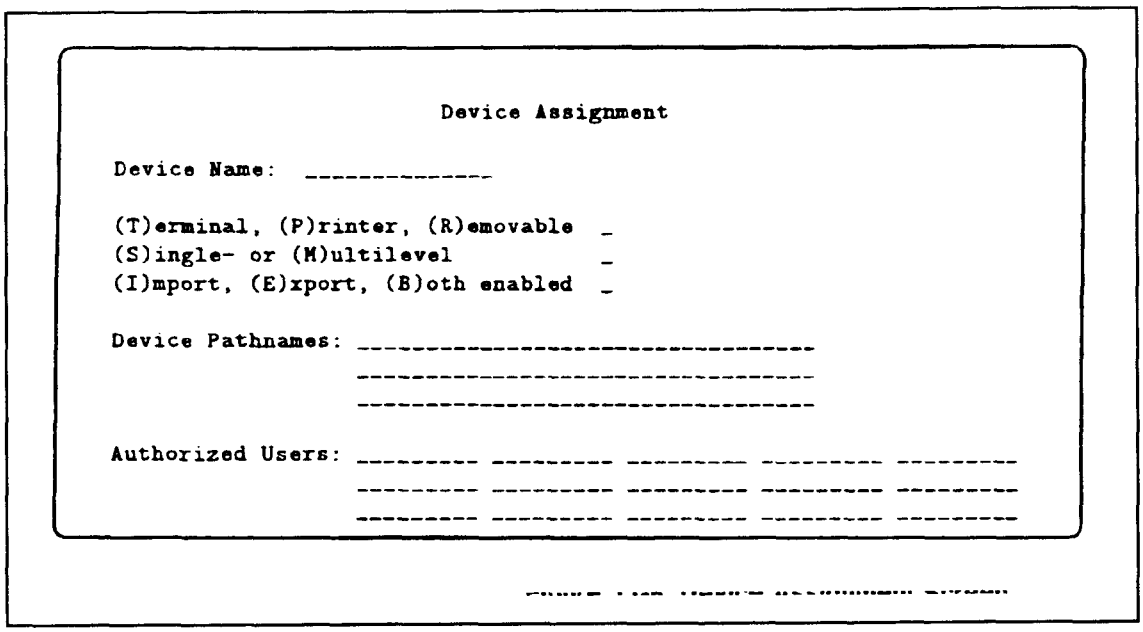

Figure 20: Device Assignment Screen [HEWL92a]

3. EL.3 Requirement

All Administrator/operator actions are auditable in they are found in the Audit System Collection Mask [HEWL92a]. The actions performed by the System Administrator, including the use of the *devasgif* command, are auditable. Therefore this requirement is met by the OS TCB component.

4. EL.4 Requirement

The TCB maintains the I/O device labels in the Device Assignment Database, a part of the HP-UX BLS operating system files. Any change to these security levels are found in the respective audit files or logs. Therefore this requirement is met by the OS TCB component.

5. EL.5 Requirement

This decomposed requirement is almost the exact requirement as stated in EL.3. The system administrator's action, if selected in the audit mask, will be audited. Thus, any

numeric label format.) Therefore this requirement is met by both the DBMS and OS TCB components.

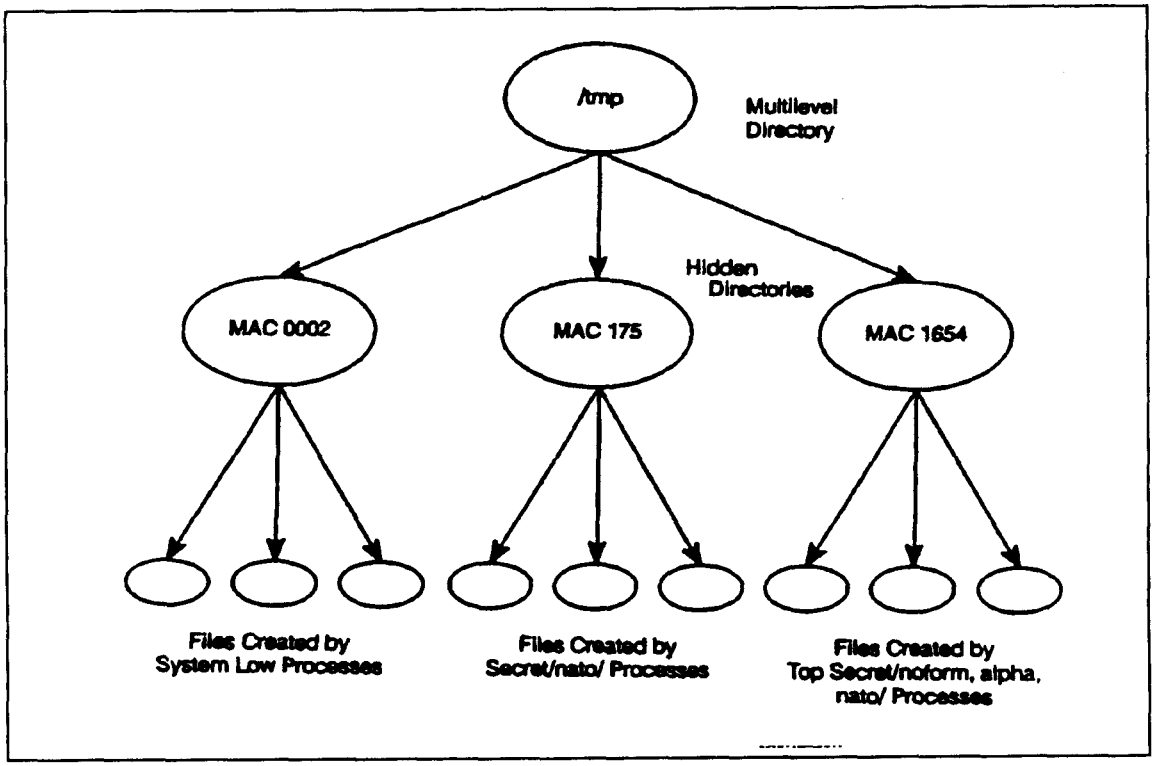

Figure 21: Hidden Directories in HP-UX BLS [HEWL92a]

2. EM.2 Requirement

The *mltape* command of HP-UX BLS correctly achieves the purpose of this requirement by copying the files into the specified device together with path name, status information, and security attributes. The security attributes of the file contain the security label. The operating system handles all copying to the output device. Therefore this requirement is met by the OS TCB component.

3. EM.i1 Requirement

The Trusted ORACLE Export utility can be used to export data out of the database by specifying the single level export option on the command line (MLS=N). If a file, created using the Export utility is later exported out of the system the *tar* and *cpio*

level of assurance [ATKI94].) Therefore this requirement is meet by the OS TCB component when the MaxSix package is installed.

6. EM.5 Requirement

This requirement is met by the MaxSix MLS network protocol. See EM.4 requirement, above. Therefore this requirement is meet by the OS TCB component.

E. EXPORTATION TO SINGLE-LEVEL DEVICES

1. ES.1 Requirement

The HP-UX BLS operating system designates (through the operating system administrator) and maintains a device in the system as either single-level or multilevel. A single level device does not store labels with the files that it outputs. However, HP-UX BLS does specify a default sensitivity level for a single level device. This means that if a single level device is labeled, then only data at that level is exported through that device. Of course no labels will be associated with the exported files. Therefore this requirement is met by default in the operating system, and a NM is placed in the DBMS TCB component column.

2. ES.2 Requirement

The Trusted ORACLE Import utility is only good for files created with the Trusted ORACLE Export utility [ORAC92a]. By default, the Import utility performs a single level import on a single level export file. The user logs into the operating system at the level at which they want the information imported into the database. It is the responsibility of the user to know (based on the export file's label or other information) at which level to import the data. Therefore, for the DBMS component of the TCB, we have labeled this requirement as "U", for user responsibility.

The *tar* and *cpio* programs of HP-UX BLS have been modified to import single-level media.These programs perform the appropriate checks against the Device Assignment database to ensure that the device used for import is in fact specified as single-

2. HRO.2 Requirement

The HP-UX BLS prints the sensitivity level of the process executing the print command on the banner page of the printout. It usually appears in the same location on the banner page as the print-job detail (i.e., filename, process number, date, etc.). As stated in HRO.1, this banner page label is the same as the user's session level.

There is no mention of marking the end of all human-readable paged, hardcopy output with a trailer page. However, the last page printed (a body page) will have a human-readable sensitivity label printed on the top and bottom of the page. Therefore, this requirement is met by the OS TCB.

3. HRO.3 Requirement

When the information is printed on the printer by the HP-UX BLS operating system, the banner page includes the sensitivity level of the process and each internal page includes the sensitivity level of the file that appears on that page. Internal pages (body pages), are labeled with the highest sensitivity level of the information that is printed on the page.

Top and bottom labeling is characteristic only of the body pages of the printout. The banner page only prints the classification one time, usually near the print-job detail. There is no mistaking a banner page with a body page in HP-UX BLS, because the banner page is uniquely designed and standard. Therefore, this requirement is met by the OS TCB.

4. HRO.4 Requirement

This requirement does not appear to be applicable in HP-UX BLS operating system. Labeling is supported on the line printer only. Labels are not supported on laser printers or plotters [HEWL92c]. In addition, labeling is not supported when the output is assigned a "landscape" orientation (i.e., the output is printed horizontally)[HEWL92c]. However, since HP-UX BLS does not support these printing options (it only prints line printer text) this requirement is not applicable to the OS TCB or the Trusted ORACLE TCB components.

requirements for details below.) Therefore, this requirement is met in both the DBMS TCB and the OS TCB.

2. MAC.2 Requirement

The Trusted ORACLE DBMS TCB component enforces a mandatory access control policy over all objects controlled by the DBMS through the labels attached to each object. This MAC policy, based on reading objects and writing objects by subjects, is discussed in depth in the MAC.6 and MAC.7 requirements below.

The HP-UX BLS TCB component maintains sensitivity labels on all objects controlled by the operating system. These objects include regular files, inter-process communication (IPC) objects, directories, special files, pipes, processes, and symbolic links. When objects are created, the HP-UX BLS system attaches security attributes to the objects. For example, when a file is created, the full pathname of the file, the file owner and group, the file mode and type, the sensitivity level, the potential and granted privilege sets, and the access control lists are stored in the File Control database. Everytime a process attempts to access the file, it must search the File Control database, check, and pass each parameter before access is granted. The MAC policy for reading and writing these objects is discussed in depth in the MAC.6 and MAC.7 requirements below. Therefore, this requirement is met in both the DBMS TCB and the OS TCB.

3. MAC.3 Requirement

Within the Trusted ORACLE TCB component, subjects and objects are assigned sensitivity levels at the time they are created. Trusted ORACLE sensitivity labels for both subjects and objects consist of four components. See Figure 22 below.

and object's sensitivity label before mandatory access in granted or denied, or if some other scheme or procedure is used to match and compare subject and object labels.

HP-UX BLS enforces the MAC policy by making sure that the user process is cleared to access information from an object by comparing the sensitivity level of the process with the sensitivity level of the object. Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

5. MAC.5 Requirement

Trusted ORACLE supports the range, size, and type of label formats provided by HP-UX BLS [ORAC92b].

HP-UX BLS has been designed to support many different configurations of sensitivity labels, with a "virtually unlimited capacity for the number of classifications and categories."[HEWL92a] The maximum classification number is set to 16 by default; the maximum category number is set to 1024 by default. Both these defaults can be changed during system installation. [HEWL92a] Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

6. MAC.6 Requirement

Trusted ORACLE provides the read operation by granting its subjects the SELECT operation. Before a database user can SELECT from an object, such as a table or view (thus reading the object), his/her label (clearance) must dominate the label of the object. The MAC rules for reading an object state, "users can read objects at their label and below; users cannot read objects at labels that they do not dominate."[ORAC92a]

The Trusted ORACLE rules above use the term "dominate". Oracle defines dominate as a relationship between labels where one label dominates another if its classification is greater than or equal to that of the other label and its categories are a superset of the other's categories (all categories are represented). This is essentially the same definition as used in the TCSEC.

The user can log on the system at any sensitivity level up to his/her clearance, which is the highest sensitivity level the user has been cleared for. (The clearance assigned to a user is determined by personnel policies, commensurate with the level of trustworthiness of the user; the Authentication Administrator sets up the user's clearance during system account creation.) In HP-UX BLS, the user's login process is tagged with the Login User ID (LUID). This indelible tag can never be changed or modified, not even by a superuser with all the system privileges. For example, even if a system administrator jumps from one user account to another (e.g., when using the *su* command), the LUID of the system administrator is inherited by the new processes spawned from the *su* program. This provides absolute accountability and always traces who did what by examination of the LUID. Therefore, this requirement is met in the OS TCB only.

9. MAC.9 Requirement

This requirement can be summarized to mean that system processes for users must be dominated by user's clearance as found in the system's I&A database. Based on available documentation, both the DBMS and the OS adhere to this requirement.

10. Additional MAC Comments

The user's LUID label (session level) is the sensitivity level that is used in Trusted ORACLE when subjects (processes) are created. No spawned process or new object created by the LUID process can exceed the sensitivity level of the LUID. The same is true when invoking special MAC privileges (WRITEDOWN, WRITEUP, READUP). When a database user connects to an Oracle database, he/she can connect at any sensitivity level up to his/her operating system clearance (I&A data).

With respect to the MAC privileges, if a database user is connected at UNCLASSIFIED and has the READUP privilege, then he or she can read higher levels of information, but only up to (and equal to) his/her overall system clearance. The READUP privilege violates the simple security property of the Bell-LaPadula model and is used as a means to prevent the user from having to log out of the system and then log back in at a

IX. INFORMIX ANALYSIS

This chapter discusses the analysis of Informix-OnLine/Secure 5.0 against the TCSEC requirements (as decomposed in Chapter VII) for labels and the mandatory access controls. The configuration that we analyzed is the one where Informix-OnLine/Secure utilizes the "raw device storage", thus circumventing most of the UNIX file system.

Again, we start by looking at the DBMS TCB component (i.e., database server software and user's manuals), then proceed to the operating system TCB. As discussed in Chapter VII, a requirement, as listed below, can be met in the DBMS TCB component, the operating system TCB component, both components, or it may not be meet or is not applicable. If user action is required to meet this requirement, then a "U" will be placed in the respective columns associated with the requirement. After each requirement, we determine where the requirement was meet, if in fact they were met at all. (See requirement summaries in Chapter VII.)

A. LABELS

1. LAB.1 Requirement

For an operating system user to gain access to Informix-OnLine/Secure, he/she must be added to a new operating system group called "ix_users." This requires the operating system administrator (OSA) to add this new group to the existing groups found in the HP-UX BLS implementation. In addition, the new database user's clearance must at least equal (i.e., dominate) the minimum clearance established for the Informix-OnLine/ Secure database (which is usually referred to as DATALO).

The database user's clearance (i.e., sensitivity label) is actually maintained within the OS tables (Protected Password database in HP-UX BLS). The Informix-OnLine/ Secure DBMS has no username tables which it actually maintains for user subjects.

they must call the LABELTOSTRING function on the LABEL attribute of the respective table.

TABLE 20: EXAMPLE OF A ROW TABLE FOR A PARTICULAR TABLE OBJECT

LABEL	ROW_ID	Data fields
10	1	
50	2	
100	3	

Note that the LABEL and ROW_ID columns in both tables above, are invisible to standard users.

A hypothetical *sysprocedures* table is shown in Figure 21, below. The *sysprocedures* table is a special table separate from the *systables* and contains only the stored procedure objects defined on the database. Because the label of the row in the *sysprocedures* table is the same as the level of the procedure to which it refers, the result of a query on the LABEL attribute returns the security level of the procedure [INFO93c].

TABLE 21: EXAMPLE OF A *SYSPROCEDURE* TABLE IN INFORMIX-ONLINE/ SECURE

LABEL	procname	pointer
10	PROC1	
50	PROC2	
100	PROC3	

The storage objects seen by the HP-UX BLS operating system (i.e., files, directories, devices, IPC objects, symbolic links, named pipes, processes, printer queues, and ptys) are maintained within the OS. Files and directories are labeled individually, and

Therefore, this requirement that the TCB shall request and receive the security level of the data, is accomplished directly by the authorized users of the system when they properly load tapes or floppy diskettes at the correctly labeled input device. This requirement is meet by the standard user, and we place a "U" in the DBMS column of the summary table, and a "NA" in the OS column.

6. LAB.5 Requirement

Standard users (not the DBSA) import unlabeled data using the *dbimport* command. The *dbimport* command is not audited, but other events associated with importing data (such as creating a new database, or locking of tables) are auditable. If a new database is not created, then this importing of non-labeled data may not be audited by Informix-OnLine/Secure. Therefore, this requirement for the auditing of non-labeled data will not have been met.

All actions performed by the DBSA and the DBSSO users are audited by default. There are no audit masks for DBSA and DBSSO users [INFO93b]. Only the DBSA can import data with OnLine/Secure internal MAC labeling from secondary storage media into an Informix-OnLine/Secure database. If the DBSA imports non-labeled data, then this requirement is met. However, if a standard user imports non-labeled data, then this action may not be audited by the system. Therefore, we place a "NM" in the DBMS column.

The HP-UX BLS operating system allows for the collection of audit data through the use of the Audit System Collection Mask. One of the audit capabilities of this system mask is the auditing of subsystem events. Since import/export of data is a subsystem event (i.e., Tape), the HP-UX BLS system is capable of auditing all events associated with the import of non-labeled data. Therefore, we place an "OS" in the HP-UX OS column of the summary tables for this requirement.

by row (under the LABEL attribute). These labels are unique to the specific implementation of the operating system; they are not necessarily the same or even understandable by other secure system implementations. Only the DBSA can export labeled data with OnLine/Secure internal MAC labeling to secondary storage media.

This requirement is met by the DBMS component and a "D" is placed in the respective column and a "NA" in the OS column.

C. EXPORTATION OF LABELED INFORMATION

1. EL.1 Requirement

The Informix-OnLine/Secure user's manual describes the desired sensitivity levels assigned to different devices as shown in Table 22:, below.

TABLE 22: SECURITY RANGES FOR DEVICES IN INFORMIX-ONLINE/ SECURE [INFO93B]

Device type	Minimum	Maximum
Terminal	Datalo	Datahi with groups IX_DATA, IX_DBSA, and IX_DBSSO (when defined)
Printer	Datalo	Datahi with groups IX_DATA, IX_DBSA, and IX_DBSSO (when defined)
Tape drives for DBSA use	Datahi, and IX_DATA	Same as minimum
Tape drives for standard users	Datalo	Datahi

However, the functions to designate each communication channel and I/O device are found within the underlying operating system, HP-UX BLS. The system administrator defines the security characteristics of each import/export device by placing the required information in the Device Assignment database. Every device that is to be used must have an entry in the Device Assignment database. Devices include terminals, line printers, and import/export devices such as tape drive systems. The operating system uses

4. EL.4 Requirement

The HP-UX BLS maintains the I/O device labels in the Device Assignment database, a part of the operating system file structure. Any change to these security levels are found in the respective audit files or logs.Therefore, this requirement is met in the OS TCB component.

5. EL.5 Requirement

This requirement is the practically the same as EL.3 All operating system administrator/operator actions are auditable in they are found in the Audit System Collection Mask [HEWL92a]. The actions performed by the System Administrator, including the use of the *devasgif* command to change security levels of devices are auditable.Therefore, this requirement is met in the OS TCB component.

6. EL.i1 Requirement

Since all operating system administrator actions are auditable (if selected in the audit system collection mask), level changes of both single level and multilevel communications channels and I/O devices will be auditable. Though this interpretation states that multilevel communications channels and I/O devices are not required to be auditable, they will be in HP-UX BLS if the system administrator actions are selected in the Audit System Collection Mask because there is no granularity in the audit mechanism. (When the system administrator actions are audited, all actions associated with the system administrator are audited.)Therefore, this requirement is met in the OS TCB component.

D. EXPORTATION TO MULTILEVEL DEVICES

1. EM.1 Requirement

Only the DBSA is capable of exporting labeled Informix-OnLine/Secure objects. The DBSA can use five functions to export labeled data: the DB-MONITOR, the *tbtape -s, tbtape -a, tbtape -c* or the *tbunload* commands. The data is written to the media

they log-on the system.) Therefore, depending on the export being made (either to tape or to printer), this requirement is met by both the OS and the DBMS TCB.

5. EM.4 Requirement

Informix-OnLine/Secure requires the use of the Informix-Star/Secure distributed client/server database product to enforce the database server's security policy to remote client workstations. The minimum requirement for a secure Class B1 configuration is MaxSix 1.0 networking software. However, client workstations can be running any network configuration, ranging from unlabeled to MaxSix 2.x, as long as the network security officer can configure the MaxSix network databases properly [INFO93f]. (For more information on MaxSix, See "EM.4 Requirement" on page 106.)

Additionally, the HP-UX BLS network transports security attributes with data and extends a host's access controls to the network subsystem so that a host can make access decisions using local policies as data traverses the network between communicating processes [HEWL92c].Therefore this requirement is meet by the OS TCB component.

6. EM.5 Requirement

This requirement is met by the MaxSix MLS network protocol. See EM.4 requirement, above. Therefore this requirement is met by the OS TCB component.

E. EXPORTATION TO SINGLE-LEVEL DEVICES

1. ES.1 Requirement

The Informix-OnLine/Secure DBMS relies upon the operating system administrator to set the device security attributes in coordination with the DBSA [INFO93b].

The HP-UX BLS operating system designates and maintains (through the operating system administrator) a device in the system as either single-level or multilevel. A single level device does not store labels with the files that it outputs. However, HP-UX BLS does specify a default sensitivity level for a single level device. This means that if a

to ensure that the device used for export is in fact specified as single-level. The Informix-OnLine/Secure *dbexport* utility must confer with the HP-UX BLS Device Assignment database before exporting single-level data. Therefore, this requirement is can be met by both the OS TCB component.

F. LABELING HUMAN-READABLE OUTPUT

Informix-OnLine/Secure provides the DB-Access utility which allows a database user to redirect the results of an SQL query (e.g., SELECT) from the screen to a printer, system file, or a program. By selecting the "Printer" option on the OUTPUT menu, the DB-Access utility sends the results of a query to the default printer. [INFO93e] The actual printing of human-readable output is handled by the HP-UX BLS operating system.

1. HRO.1 Requirement

The operating system administrator only indirectly specifies the printable label names associated with exported security labels. Labels names are set-up when the sensitivity classifications and categories are defined for the system. Whenever objects are created (either database objects or operating system objects), they are labeled with the creating process's sensitivity level.

HP-UX BLS prints the sensitivity label of the process producing the output on the banner page of the printout. This is the sensitivity label that was typed when the user logged into the system. Therefore, this requirement is met by the OS TCB component.

2. HRO.2 Requirement

The HP-UX BLS prints the sensitivity label of the process producing the output on the banner page of the printout. It usually appears in the same location on the banner page as the print-job detail. As stated in HRO.1, this is the sensitivity label that was typed when the user logged into the system.

There is no mention of marking the end of all human-readable paged, hardcopy output with a trailer page. However, the last page printed (a body page) will have a human-

which removes the filtering done on printed output. Both of these options are audited by the system. [HEWL92a]

As a side note, both the special options to the *lp* command suppress the internal printout labeling features of HP-UX BLS. The assumption is that the banner page is still printed. Therefore, this requirement is met by the OS TCB component.

G. MANDATORY ACCESS CONTROL

1. MAC.1 Requirement

Subjects in Informix-OnLine/Secure are operating system processes that use On-Line/Secure. The DBMS server mandates that every access to every piece of data (i.e., object) by all users (i.e., subjects) be checked to see if access is permissible, based on the sensitivity label of the data and clearance of the user. Informix-OnLine/Secure uses the same set of sensitivity labels that are available in HP-UX BLS. Therefore, there is no problem with comparison of subject and object labels when the DBMS assigns labels to objects based on the subject's sensitivity level. (Specific access rules will be discussed in MAC.6 and MAC.7 requirements.)

The HP-UX BLS TCB component maintains sensitivity labels on all subjects (active entities in the system, such as processes). When a user logs into the system, a login user ID (LUID) is attached to the user's login process. The LUID points to the Protected Password database which contains the clearance level for that particular subject. (The clearance level is the highest sensitivity label allowed for that subject's process). All processes spawned from this LUID process contain this same LUID with its associated clearance level. From these subject processes' labels a mandatory access control policy, based on the Bell-LaPadula security model is enforced. (See the MAC.6 and MAC.7 requirements for details below.) Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

SECRET, called an access level, and zero or more categories, such as CRYPTO, NATO, and PROPRIETARY[INFO93c]. Informix-OnLine/Secure uses the same set of sensitivity labels that are available in HP-UX BLS. (The DBSA sets up labels in the DBMS to equal those in the OS.) Therefore, there is no problem with comparison (i.e., label equality, label dominance, etc.) of subject and object labels when the DBMS assigns labels to objects based on the subject's sensitivity level.

The HP-UX BLS system specifically supports the US DOD method of classifying information according to hierarchial classification levels and non-hierarchial categories. Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

4. MAC.4 Requirement

Subjects can access objects in Informix-OnLine/Secure by comparing sensitivity labels (in their integer tag format) of objects and subjects.

HP-UX BLS enforces the MAC policy by making sure that the user process is cleared to access information from an object by comparing the sensitivity level of the process with the sensitivity level of the object. Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

5. MAC.5 Requirement

Informix-OnLine/Secure uses the same set of sensitivity labels that are available in the operating system [INFO93c]. (See below.)

HP-UX BLS has been designed to support many different configurations of sensitivity labels, with a "virtually unlimited capacity for the number of classifications and categories."[HEWL92a] The maximum classification number is set to 16 by default; the maximum category number is set to 1024 by default. Both these defaults can be changed during system installation. [HEWL92a] Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

The user can log on the system at any sensitivity level up to his/her clearance, which is the highest sensitivity level the user has been cleared for. (This clearance is established by the security policy outside the automated system environment and the Authentication Administrator sets it up during system account creation). In HP-UX BLS, the user's login process is tagged with the Login User ID (LUID). This indelible tag can never be changed or modified, not even by a superuser with all the system privileges. For example, even if a system administrator jumps from one user account to another (e.g., when using the su command), the LUID is still inherited by the new processes spawned from changes accounts. This provides absolute accountability and always traces who did what by examination of the LUID. Therefore, this requirement is met in the OS TCB only.

9. MAC.9 Requirement

This requirement can be summarized to mean that system processes for users must be dominated by user's clearance as found in the system's I&A database. Based on available documentation, both the DBMS and the OS adhere to this requirement.

10. Additional MAC Comments

Additionally, in Informix-OnLine/Secure, trusted subjects or processes are created when invoking discrete privileges (PRIV_CANSETLEVEL and PRIV_CANSETIDENTITY). The PRIV_CANSETLEVEL allows a user to alter the session sensitivity level of the current session by invoking the SET SESSION LEVEL statement. The user can operate only at sensitivity levels that are dominated by the level of his/her original login session. The SET SESSION LEVEL allows the user, (by creating temporary tables and then changing session level), to change the object levels. This is a violation of the tranquility property of Bell-Lapadula model.

The PRIV_CANSETIDENTITY discrete privilege allows users to circumvent the DAC protection for database objects by adopting the user name of any Informix-OnLine/Secure nonadministrative user. The user's login ID and sensitivity level still determine what level of data the user holding this privilege can access.

X. COMPARISON OF TRUSTED ORACLE AND INFORMIX

In this chapter we will list both the similarities and the differences of Trusted ORACLE 7 and Informix-OnLine/Secure 5.0 in the context of the TCSEC requirements examined in the two previous chapters.

A. LABELS

1. LAB.1 Requirement

Trusted ORACLE maintains an ALL_USERS table for storing the database (user) subjects, by keeping a user's name, clearance, and security profile. Informix-OnLine/Secure has no such comparable table, and exclusively uses the HP-UX BLS operating system Protected Password database to maintain its database users and their security clearances. Informix requires that three new groups be added to the HP-UX BLS groups: ix_data, ix_dbsso, and ix_dbsa.

2. LAB.2 Requirement

Trusted ORACLE 7.0 and Informix-OnLine/Secure 5.0 both maintain their data dictionaries by defining objects as rows in a system table. Each row (which defines a specific object) is labeled with the sensitivity level of the subject that created it and thus is the objects label. The objects in each system are similar in nature, as shown in Table 23, below. An "*" placed in a column means that specific product does not have a comparable object in the other product.

3. LAB.i1 Requirement

There are no significant differences between Trusted ORACLE 7.0 and Informix-OnLine/Secure 5.0 with respect to this requirement.

4. LAB.3 Requirement

There are no significant differences between Trusted ORACLE 7.0 and Informix-OnLine/Secure 5.0 with respect to this requirement. Both database server systems use labeled objects and subjects with an extended version of the Bell-LaPadula model to control mandatory access between subjects to objects within the system.

5. LAB.4 Requirement

Trusted ORACLE uses the Import utility to import a single level (or multilevel) OS file into the database. Informix utilizes the *dbimport* command to do this function. Users are responsible to ensure that their session levels match the sensitivity of the data being imported.

6. LAB.5 Requirement

When importing OS files into their respective databases, neither Oracle nor Informix audits the import utilities or commands. When the HP-UX BLS OS imports files into the system (from outside the system), these actions are audited.

B. LABEL INTEGRITY

1. LI.1 Requirement

Trusted ORACLE stores username accounts and schema objects as rows in a system table within the data dictionary and sensitivity labels are associated with these rows. Informix-OnLine/Secure stores only schema objects, as rows in a system table and labels each row with the objects sensitivity label. Label integrity is maintained by the TCB.

tbunload command, to send multilevel data directly to an output device, using the *-t* device option.

3. EM.i1 Requirement

There are no substantial differences between Oracle and Informix with respect to this requirement.

4. EM.3 Requirement

Trusted ORACLE sends labeled objects to I/O devices with their labels as either a MLSLABEL or RAW MLSLABEL data types. Informix sends labeled objects with labels as 4-byte integer tags.

5. EM.4 Requirement and EM.5 Requirement

The MaxSix secure networking package is required for Trusted ORACLE and Informix-OnLine/Secure. We believe that both products meet this requirement by utilizing the MaxSix protocol.

E. EXPORTATION TO SINGLE-LEVEL DEVICES

There are no substantial differences between Trusted ORACLE and Informix with respect to the requirements ES.1 - ES.3.

F. LABELING OF HUMAN-READABLE OUTPUT

All requirements decomposed under this heading are met in the same way by both Trusted ORACLE and Informix-OnLine/Secure, because both DBMSs rely on the underlying operating system, HP-UX BLS exclusively to print human-readable output to the line printers. We found no mention that sensitivity labels are printed to the terminal screens.

G. MANDATORY ACCESS CONTROLS

With respect to the decomposed requirements, MAC.1 - MAC.9, there are no substantial differences between Trusted ORACLE and Informix-OnLine/Secure. The

privilege, which is allowed in the BLP model (but not in the extended-BLP which is the MAC model used in the three products), can be accomplished in the products by granting the appropriate special privileges, as indicated in Table 24.

The PRIV_CANSETLEVEL discrete privilege in Informix-OnLine/Secure, allows a database user to toggle back and forth between session levels without logging out and logging back into the system. Trusted ORACLE can accomplish this function by granting a user the MAC privileges (i.e., WRITEDOWN, READUP, WRITEUP).

The PRIV_CANSETIDENTITY in Informix-OnLine/Secure (not shown in Table 24 because, to our knowledge, Trusted ORACLE has no equivalent privilege) allows a database user to assume the identity of another database user, thus having access to that database user's owned objects. This is a way for Informix to bypass discretionary access controls, without becoming a special user, such as the DBSA or the DBSSO. Additionally, we do not know which privilege in the underlying operating system, HP-UX BLS, coincides with the PRIV_CANSETIDENTITY Informix privilege. The only user in the operating system, who can bypass discretionary access controls on unowned objects, is usually the *root* account (i.e., superuser). Therefore, we can only assume, that when a database user is granted the PRIV_CANSETIDENTITY, they have a superuser account.

the Bell-LaPadula security model (i.e., subjects writing to objects must have equal sensitivity levels). Objects are stored in data files which can be stored in raw devices of the system controlled by the Oracle database server.

Trusted ORACLE does employ trusted subjects when a user executes the MAC privilege WRITEDOWN. The WRITEDOWN MAC privilege clearly violates the Bell-Lapadula model (*-property), and is truly a trusted subject as defined by [GASS88]. The WRITEDOWN privilege is utilized when conducting a full database import and a high-level (i.e, system high) process is allowed to write data to the database at its actual label, which can be lower than system high level.

The READUP MAC privilege of Trusted ORACLE is used by database users to read objects at higher levels than their session level. (This privilege violates the simple security property of the Bell-LaPadula model.) However, this privilege is limited to reading objects with a sensitivity label dominated by the overall clearance of the user, as defined in the operating system's Protected Password database. The READUP privilege only overrides the user's session sensitivity level, not the user's system clearance level.

In addition, the CREATE TABLE AS command allows a standard user with no MAC privileges to change his/her table(s) from one sensitivity label to another (up to their DBMS sensitivity label) [ORAC92a].This is actually done by creating a new table (with all the same attributes as the old table) at a new sensitivity level, and then copying all data from the old table into the new table. The new table's definition is the exact same as the old table, just with a new sensitivity label. The old table is then dropped from the database.

Any user in Trusted ORACLE can change the labels of his/her rows within their owned tables at any time. This raises a serious concern about the persistence of row and table object labels. The Bell-LaPadula Model implies the tranquility contraint that object access classes cannot change; object labels must remain unchanged through their lifetime. Trusted ORACLE circumvents this tranquility restriction, carefully (and apparently purposely), by not defining rows as objects. Instead, they call the changing of a row's ROWLABEL pseudo-column, the "reclassifying of data." [ORAC92a] To reclassify data

3. Limitations of Research

A thorough, technical analysis was not possible with only public documentation. We acquired the following types of documentation for our research:

- Technical Overviews and Briefs
- Trusted Facility Manuals
- Administrator User's Guides
- Security Features User's Guides
- other public documents

These types of documents are insufficient to determine if products can meet the TCSEC requirements. These types of manuals and guides do not focus on the security features as they relate to the TCSEC and there is little or no discussion on the design of the system's security mechanisms. (The organization of the documents do not coincide, structurally with the TCSEC.)

B. RECOMMENDATIONS

1. MAC Downgrade Policy

All system administrators of a Trusted ORACLE or an Informix-OnLine/Secure database should develop a security policy for downgrading object labels. We recommend that only the system security officer in conjunction with the DBA be allowed to make object level changes, following a security policy for reclassifying data.(Separation of privilege would require both administrators to agree on object label changes before an actual change can be made in the system.) Specific rules with respect to this policy, cannot be defined beforehand, but a generic policy addressing the labeling and downgrading of objects should be established. This will allow users some notion of how to label objects and how to reclassify objects once they are defined. All reclassifying of data should be audited by default with no override capabilities.

C. FUTURE RESEARCH

Future research in this area could lead to a comprehensive evaluation handbook for the analysis of DBMSs. Upon the release of the Final Evaluation Reports by the NCSC for Trusted ORACLE and Informix-OnLine/Secure, study can continue to correlate the findings by NCSC with the findings of this research. Our analysis method can be expanded to cover all the TCSEC requirements, not just the MAC and labeling requirements.

A number of research questions have been posited over the years concerning the evaluation of software products which are "layered" or placed on top of trusted (previously evaluated) operating systems, such as DBMSs. Such questions are:

- Can a DBMS be evaluated like an operating system? An operating system manages resources, whereas a DBMS manages information.What is the relationship?
- Are TCB subset architectures, in the context of DBMSs, amenable to incremental evaluation (i.e., evaluating the application only without evaluating the underlying TCB subset [typically the operating system], which has already been successfully evaluated).[CHOK92] The basis for this question is the hope that the underlying operating system would not have to be reevaluated in order to evaluate the DBMS software product.
- Can or should the TCSEC, which was written initially for operating systems and their evaluations, be used for the evaluations of other systems, such as DBMS? Some authors have argued, that different criteria (criteria other than the TCSEC) should be developed for DBMS products. [GRAU90]

Future research in these areas can help determine the answers to these questions.

[DOWN94] Downs, Deborah, Chief Evaluator, Aerospace Corp., Los Angeles, CA.,Personal conversation via e-mail, 3 May 1994.

[DOYL91] Doyle, Sean, *An Introduction: Trusted ORACLE RDBMS,* Oracle Corporation, Redwood City, CA, February 1991.

[EDGE93] Edge Publishing, *EDGE: Workgroup Computing Report*, Version 4, Number 163, Edge Publishing Inc., July 5, 1993.

[EHRS91] Ehrsam, Tim, and Doyle, Sean, *ORACLE and Secure Systems: Questions and Answers*, Oracle Corporation, Redwood City, CA, October 1991.

[ENDO93] Endoso, Joyce, "DOD wants to Dump Mil Specs and Use More COTS Products", *Government Computer News*, Volume 12, Number 13, Page 3, June 21, 1993.

[GALL94] Gallagher, Sean, "Trusted ORACLE7 database is on brink of B1 security rating," *Government Computer News*, Volume 13, Number 4, February 21, 1994.

[GASS88] Gasser, Morrie, *Building a Secure Computer System*, Van Nostrand Reinhold, 1988.

[GRAU90] Graubart, Richard, "Comparing DBMS and Operating System Security Requirements: The Need for a Separate DBMS Security Criteria," *Database Security, III: Status and Prospects*, North-Holland, 1990.

[HALE94] Hale, Mike, Interview with Mike Hale at NCSC, Ft Meade, MD., 5 April 1994.

[HEWL92a] Hewlett-Packard, *HP-UX B-Level Security Trusted Facility Manual*, Edition 1, HP 9000 Computers, Hewlett-Packard Company, Palo Alto, CA, February 1992.

[HEWL92b] Hewlett-Packard, *HP-UX B-Level Security User's Guide*, Edition 1, HP 9000 Computers, Hewlett-Packard Company, Palo Alto, CA, February 1992.

[HEWL92c] Hewlett-Packard, *B1 Networking Security Features User's Guide*, First Edition, HP 9000 Computers, Hewlett-Packard Company, Palo Alto, CA, January 1992.

[HINK90] Hinke, Thomas H., "DBMS Trusted Computing Base Taxonomy," *Database Security, III: Status and Prospects*, North-Holland, 1990.

[NCSC92a] National Computer Security Center, *The Design and Evaluation of INFOSEC Systems: The Computer Security Contribution to the Decomposition Discussion,* C Technical Report 32-92, June 1992.

[NCSC92b] National Computer Security Center, *A Guide to Understanding Object Reuse in Trusted Systems,* NCSC-TG-018, Version 1, July 1992.

[NCSC92c] National Computer Security Center, *Guidelines for Writing Trusted Facility Manuals,* NCSC-TG-016, Version 1, October 1992.

[ORAC92a] Oracle Corporation, *Trusted ORACLE Administrator's Guide*, Version 1.0, Oracle Corporation, Redwood City CA, 1992.

[ORAC92b] Oracle Corporation, *ORACLE for HP-UX BLS Installation and User's Guide*, Oracle Version 7.0.9 (Developer's Release), Trusted ORACLE Version 1.0, Oracle Corporation, Redwood City CA, June 5, 1992.

[ORAC92c] Oracle Corporation, *ORACLE RDBMS: Database Administrator's Guide*, Volume 1, Version 7.0, Oracle Corporation, Redwood City, CA 1992.

[ORAC92d] Oracle Corporation, *ORACLE RDBMS: Database Administrator's Guide*, Volume 2, Version 7.0, Oracle Corporation, Redwood City, CA 1992.

[ORAC92e] Oracle Corporation, *ORACLE RDBMS: Database Administrator's Guide*, Volume 3, Version 7.0, Oracle Corporation, Redwood City, CA 1992.

[PFLE89] Pfleeger, Charles, P., *Security in Computing*, PTR Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.

[SALE93] Salemi, Joe, *Client/Server Computing with ORACLE,* Ziff-Davis Press, Emeryville, CA, 1993.

[SALT75] Saltzer, J.H., Schroeder, M.D., *The Protection of Information in Computer Systems,* IEEE, 1975.

[SCHA84] Schaefer, Marvin and Schell, Roger, "Toward an Understanding of Extensible Architectures for Evaluated Trusted Computer System Products", *Proceedings of the 1984 Symposium on Security and Privacy,* April 1984.

[SCHA91] Schaefer, Marvin, *Reflexions on Current Issues in Trusted DBMS*, Database Security IV: Status and Prospects, North-Holland 1991.

[SHOC87] Shockley, W.R. and Schell, Roger, "TCB Subsetting for Incremental Evaluation", *Proceeding of the Third AIAA Conference on Computer Security*, December 1987.